# RaceMate Reference

Welkin Systems Limited

# RaceMate

## horse racing statistics and analysis software

***by Welkin Systems Limited***

*RaceMate is a powerful race data analysis tool. Its huge database provides you with all necessary data for your betting analysis. Its versatile features help you effectively find out investment hints and improve chances of winning.*

*You will be much impressed by the lightning speed of RaceMate because its data is driven by the World's fastest desktop database engine Microsoft Visual FoxPro. Since its database resides in your local hard disk, there is no need to dedicate an Internet connection while you are using the software. You may install RaceMate in a notebook and use it anytime and anywhere.*

# Table of Contents


## Part I Introduction to RaceMate 18

1 Main Features ... 18
2 System Requirements ... 20
3 Installing RaceMate ... 21
4 Registering and Activating RaceMate ... 23
5 Installing into More Than One Computer ... 28
6 About Upgrading Windows ... 32
7 Remarks about Inputting Chinese ... 33

## Part II Release Notes 36

1 What is New in RaceMate 11.5 ... 36
2 What is New in RaceMate 10.5 ... 37
3 What is New in RaceMate 10.0 ... 38
4 What is New in RaceMate 9.8 ... 40
5 What is New in RaceMate 9.5 ... 43
6 What is New in RaceMate 9.0 ... 47
7 What is New in RaceMate 8.0 ... 50
8 What is New in RaceMate 7.0 ... 59

## Part III Data Update 66

1 Connecting to the Internet ... 66
2 Dial-Up Connection ... 67
3 Downloading Data ... 69
4 Repeat Last Download ... 69
5 Firewall Blocking ... 70

## Part IV Basic UI Techniques 72

1 Using Menus and Toolbars ... 72
- Menus and Toolbars ... 72
- Moving Toolbars ... 72

2 Changing Language of Display ... 72
3 Hyper-Linking Records ... 73
4 Incremental Search ... 74
5 Editing Data ... 77
6 Handling Forms ... 77
- Multiple Instances of a Form ... 78
- Freezing a Form ... 79
- Changing and Restoring Form Size ... 79

7 Handling Grids ... 79
- Changing Column Width ... 80


© Welkin Systems Limited

Changing Row Height ... 81
Freezing Columns ... 82
Changing a Column's Position ... 83
Sorting Grid Columns ... 84
Resetting Grid Layout ... 86

Part V Main Data Forms 90

1 Record Tree Form ... 90
2 Race Calendar Form ... 94
3 Race Data Form ... 95
Starter View Page (Race Data Form) ... 96
Trackwork Page (Race Data Form) ... 97
Allocation Page (Race Data Form) ... 98
Flemming Page (Race Data Form) ... 99
Dividends Page (Race Data Form) ... 100
Remark Page (Race Data Form) ... 100
Links Page (Race Data Form) ... 101
Index Page (Race Data Form) ... 101
Print Race Data (Race Data Form) ... 102
4 Race Search Form ... 103
5 Jockey Silks ... 104
6 Horse Form ... 104
Particulars Page (Horse Form) ... 105
Pedigree Page (Horse Form) ... 106
Past Record and Comprehensive Statistics for (Horse Form) ... 106
Overseas Page (Horse Form) ... 107
Wins Page (Horse Form) ... 107
Trackwork (Horse Form) ... 108
Links Page (Horse Form) ... 108
Horse List Page (Horse Form) ... 109
7 Jockey Form ... 109
Past Record and Comprehensive Statistics (Jockey Form) ... 109
Wins Page (Jockey Form) ... 110
Particulars/Notes Page (Jockey Form) ... 111
Links Page (Jockey Form) ... 111
Jockeys List Page (Jockey Form) ... 112
8 Stable Form ... 112
Horses in Stable (Stable Form) ... 113
Wins Page (Stable Form) ... 113
Past Record and Comprehensive Statistics (Stable Form) ... 113
Notes Page (Stable Form) ... 114
Links Page (Stable Form) ... 114
Stables List Page (Stable Form) ... 115
9 Current Odds from Jockey Club ... 115
10 Daily Track Work and Barrier Trial ... 115
Barrier Trials (Daily Track Work Form) ... 116
Barrier Trial Groups (Daily Track Work Form) ... 116
Track Work (Daily Track Work Form) ... 116
Links Page (Daily Track Work Form) ... 117
11 Jump by Double-Clicking a Column ... 117
Defining Double-Click Actions ... 117
Forms and Record IDs ... 119
12 Comprehensive Statistics and Record Query Form ... 119

© Welkin Systems Limited

# Part VI Views Management 122

1 Type 1: Starter Views ... 122

Defining Starter Views ... 122
Creating Starter Views ... 123
Modifying Existing Views ... 127
View Header ... 128
Example: Creating a Starter View ... 130
Advanced Settings of View Columns ... 135
Filtering Past Records ... 136
Rearranging Position of Views in View List ... 136
Exporting and Importing View Structures ... 137
Copying a Starter View ... 139
Show History in Starters Views ... 139
History Tracks ... 139
Sorting and Grouping ... 141
Sorting in Individual History Groups ... 141
Setting Values & Choosing MCL Variables ... 142
Filtering Records in Starters Views ... 142
Selecting Criteria ... 142
Modifying Criteria and Setting Compound Filters ... 143
Filter Editor ... 146

2 Type 2: History and Statistics Views ... 148

Types of History and Statistics Views ... 148
Argument of History/Stat View ... 149
Setting Data Range in an Argument ... 149

3 Cache for View Data ... 149

4 Conversion Tables ... 150

Managing Conversion Tables ... 151
Using Conversion Tables in Views ... 153

5 Linking Up Views ... 155

Linked Columns ... 155

6 Exporting and Importing View Data ... 158

# Part VII MCL Variables in MCL Functions 160

1 Introduction to MCL Variables ... 160

2 Setting Values & Choosing MCL Variables ... 161

3 Using MCL Variables As View Arguments ... 162

4 MCL Variables to Set Data Range ... 164

# Part VIII Built-in Views 168

1 Built-in Starter Views ... 168

All Distances WQP Stats View ... 168
Jockey Change Statistics ... 171

# Part IX Excel Spreadsheet Features 178

1 How to Create Complicated Formulae ... 178

2 MCL Debugging ... 178

# Part X Printing and Creating Reports 182

1 Creating a Report for a View ... 182


© Welkin Systems Limited

2 Report Designer ... 183

Part XI Multimedia Resources 188

1 Autolinks ... 188
2 Managing Your Own Multimedia Resources ... 189
3 Saving Web Video/Photo into Disk ... 191

Part XII Statistics and Data Analysis 196

1 Type 1: Comprehensive Statistics & Record Query ... 196
Defining Structure of Statistics Views and Queries ... 197
Statistical View and Record Filtering Results ... 201
Editing View Particulars ... 202
Editing View Column/Criteria Items ... 204
Query Operators for Requirements ... 206
Creating and Removing OR or AND Criteria Group ... 209
Data Period and Display Mode ... 211
Statistical Items - the [Stat] Column ... 214
Viewing Past Records and Move Columns in Stat View Page ... 216
2 Type 2: Statistics via Starter Views (Race Data Form) ... 218
Betting Index ... 218
Evaluating Accuracy of Prediction Models ... 218
Example: Draw Position Statistics ... 219
Basic Steps ... 220
Statistical Groups ... 222
Printing or Exporting Statistics ... 224
3 Type 3: Simplified Statistics Form ... 225

Part XIII Exporting and Importing Data 228

1 Exporting Simple Race Data ... 228
2 Exporting Horse Data ... 229
3 Exporting or Copying Grids Cells ... 230
4 Exporting View Data ... 230
5 Data Export via Starter Views ... 233
6 Format of Exported Excel Files ... 238
7 Importing View Data From Excel Files ... 240

Part XIV System Maintenance and Configuration 244

1 Backing Up and Restoring User Data ... 244
2 RaceMate Notice ... 245
3 System Configurations and Information ... 246
Options and System Information ... 246
About RaceMate ... 247
4 Clearing Cache Folder ... 248
5 Upgrading RaceMate or Installing Add-Ins ... 248

Part XV MCL Tutorial 250

Part XVI MCL Reference 252

© Welkin Systems Limited

**1 Arguments and Variables of MCL functions ... 253**

Optional Arguments ... 253
Data Types of Arguments ... 253
The nBackRaces Argument of an MCL function ... 254
The fCondition Argument (MCL Condition) ... 254
When to use the While keyw ord ... 255
Quoted MCL Expression Arguments ... 255
The DefaultVal Argument ... 256
The cIncluding Argument ... 256

**2 Null Values ... 257**

Characteristics of Null Values ... 257
Absence of Values ... 257
Null Values in Statistics ... 257
Propagation of Null Values ... 257
Errors in MCL Formulas ... 258

**3 Information about a Start ... 258**

HrAge( ), JrAge( ), SrAge( ) ... 259
HrAgeMonths( ), JrAgeMonths( ), SrAgeMonths( ) ... 259
HrAllow ance( ), JrAllow ance( ), SrAllow ance( ) ... 260
HrBleedDate( ), JrBleedDate( ), SrBleedDate( ) ... 260
HrBleedDateStr( ), JrBleedDateStr( ), SrBleedDateStr( ) ... 260
HrBlinker( ), JrBlinker( ), SrBlinker( ) ... 260
HrBlinkerAbbr( ), JrBlinkerAbbr( ), SrBlinkerAbbr( ) ... 261
HrBlinkerAbbrNew ( ), JrBlinkerAbbrNew ( ), SrBlinkerAbbrNew ( ) ... 261
HrBlinkerNew ( ), JrBlinkerNew ( ), SrBlinkerNew ( ) ... 261
HrBodyWeight( ), JrBodyWeight( ), SrBodyWeight( ) ... 262
HrClass( ), JrClass( ), SrClass( ) ... 262
HrClassID( ), JrClassID( ), SrClassID( ) ... 262
HrCurRating( ), JrCurRating( ), SrCurRating( ) ... 263
HrDraw ( ), JrDraw ( ), SrDraw ( ) ... 263
HrEntriesWeight( ), JrEntriesWeight( ), SrEntriesWeight( ) ... 264
HrFavorite( ), JrFavorite( ), SrFavorite( ) ... 264
HrFinalOdds( ), JrFinalOdds( ), SrFinalOdds( ) ... 265
HrFP( ), JrFP( ), SrFP( ) ... 265
HrFPX( ), JrFPX( ), SrFPX( ) ... 266
HrHorseNo( ), JrHorseNo( ), SrHorseNo( ) ... 266
HrHorseNoStr( ), JrHorseNoStr( ), SrHorseNoStr( ) ... 266
HrIsRunner( ), JrIsRunner( ), SrIsRunner( ) ... 267
HrLastSectPos( ), JrLastSectPos( ), SrLastSectPos( ) ... 267
HrLoadDiff( ), JrLoadDiff( ), SrLoadDiff( ) ... 268
HrLoadDiffStr( ), JrLoadDiffStr( ), SrLoadDiffStr( ) ... 268
HrMargin( ), JrMargin( ), SrMargin( ) ... 268
HrMarginLen( ), JrMarginLen( ), SrMarginLen( ) ... 269
HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( ) ... 269
HrNetLoadDiff( ), JrNetLoadDiff( ), SrNetLoadDiff( ) ... 270
HrOdds( ), JrOdds( ), SrOdds( ) ... 270
HrOdds1Day( ), JrOdds1Day( ), SrOdds1Day( ) ... 270
HrOdds5Min( ), JrOdds5Min( ), SrOdds5Min( ) ... 271
HrOverw eight( ), JrOverw eight( ), SrOverw eight( ) ... 271
HrPastIncident( ), JrPastIncident( ), HrPastIncident( ) ... 271
HrPlaceTickets( ), JrPlaceTickets( ), SrPlaceTickets( ) ... 272
HrPositions( ), JrPositions( ), SrPositions( ) ... 272
HrPrivilege( ), JrPrivilege( ), SrPrivilege( ) ... 272
HrPrivilegeID( ), JrPrivilegeID( ), SrPrivilegeID( ) ... 273
HrRunning( ), JrRunning( ), SrRunning( ) ... 273
HrRacingWeight( ), JrRacingWeight( ), SrRacingWeight( ) ... 273
HrRating( ), JrRating( ), SrRating( ) ... 274


© Welkin Systems Limited

HrRatingDiff( ), JrRatingDiff( ), SrRatingDiff( ) ... 274
HrReserved( ), JrReserved( ), SrReserved( ) ... 274
HrSeconds( ), JrSeconds( ), SrSeconds( ) ... 275
HrSectionPos( ), JrSectionPos( ), SrSectionPos( ) ... 275
HrSectMargins( ), JrSectMargins( ), SrSectMargins( ) ... 275
HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( ) ... 276
HrSectMargLensX( ), JrSectMargLensX( ), SrSectMargLensX( ) ... 277
HrSectPositions( ), JrSectPositions( ), SrSectPositions( ) ... 277
HrSectTime( ), JrSectTime( ), SrSectTime( ) ... 278
HrSectTimew ( ), JrSectTimew ( ), SrSectTimew ( ) ... 279
HrSpecialBit( ), JrSpecialBit( ), SrSpecialBit( ) ... 279
HrSpecialBitAbbr( ), JrSpecialBitAbbr( ), SrSpecialBitAbbr( ) ... 279
HrSpeed( ), JrSpeed( ), SrSpeed( ) ... 280
HrStablePrefence( ), JrStablePrefence( ), SrStablePrefence( ) ... 280
HrStartID( ), JrStartID( ), SrStartID( ) ... 280
HrStartIDOn(), JrStartIDOn(), SrStartIDOn() ... 281
HrStakes( ), JrStakes( ), SrStakes( ) ... 281
HrTime( ), JrTime( ), SrTime( ) ... 283
HrTimew ( ), JrTimew ( ), SrTimew ( ) ... 283
HrUpdow n( ), JrUpdow n( ), SrUpdow n( ) ... 283
HrUpdow nID( ), JrUpdow nID( ), SrUpdow nID( ) ... 284
HrWeight( ), JrWeight( ), SrWeight( ) ... 284
HrWinTickets( ), JrWinTickets( ), SrWinTickets( ) ... 285
HrWQPStat( ), JrWQPStat( ), SrWQPStat( ) ... 285

**4 Static information of Horse, Jockey, and Stable ... 285**

HrArrivalYear( ), JrArrivalYear( ), SrArrivalYear( ) ... 286
HrBrandNo( ), JrBrandNo( ), SrBrandNo( ) ... 287
HrBreedNote( ), JrBreedNote( ), SrBreedNote( ) ... 287
HrCHorse( ), JrCHorse( ), SrCHorse( ) ... 287
HrColor( ), JrColor( ), SrColor( ) ... 288
HrCountryID( ), JrCountryID( ), SrCountryID( ) ... 288
HrDam( ), JrDam( ), SrDam( ) ... 288
HrDDam( ), JrDDam( ), SrDDam( ) ... 289
HrDDDam( ), JrDDDam( ), SrDDDam( ) ... 289
HrDDSire( ), JrDDSire( ), SrDDSire( ) ... 289
HrDSDam( ), JrDSDam( ), SrDSDam( ) ... 290
HrDSire( ), JrDSire( ), SrDSire( ) ... 290
HrDSSire( ), HrDSSire( ), SrDSSire( ) ... 291
HrEHorse( ), JrEHorse( ), SrEHorse( ) ... 291
HrFoalDate( ), JrFoalDate( ), SrFoalDate( ) ... 291
HrHorse( ), JrHorse( ), SrHorse( ) ... 292
HrHorseID( ), JrHorseID( ), SrHorseID( ) ... 292
HrHorseMarkNote( ), JrHorseMarkNote( ), SrHorseMarkNote( ) ... 292
HrHorseNote( ), JrHorseNote( ), SrHorseNote( ) ... 293
HrJockey( ), JrJockey( ), SrJockey( ) ... 293
HrJockeyClass( ), JrJockeyClass( ), SrJockeyClass( ) ... 293
HrJockeyEngAbbr( ), JrJockeyEngAbbr( ), SrJockeyEngAbbr( ) ... 294
HrJockeyID( ), JrJockeyID( ), SrJockeyID( ) ... 294
HrJockeyRetired( ), JrJockeyRetired( ), SrJockeyRetired( ) ... 294
HrNoteMark( ), JrNoteMark( ), SrNoteMark( ) ... 295
HrOrigin( ), JrOrigin( ), SrOrigin( ) ... 295
HrOriginCategory( ), JrOriginCategory( ), SrOriginCategory( ) ... 295
HrOw ner( ), JrOw ner( ), SrOw ner( ) ... 296
HrRetired( ), JrRetired( ), SrRetired( ) ... 296
HrSameBlood( ), JrSameBlood( ), SrSameBlood( ) ... 296
HrSDam( ), JrSDam( ), SrSDam( ) ... 297
HrSDDam( ), JrSDDam( ), SrSDDam( ) ... 297
HrSDSire( ), JrSDSire( ), SrSDSire( ) ... 298

© Welkin Systems Limited

HrSex( ), JrSex( ), SrSex( ) ... 298
HrSickHistory( ), JrSickHistory( ), SrSickHistory( ) ... 298
HrSire( ), JrSire( ), SrSire( ) ... 300
HrSSDam( ), JrSSDam( ), SrSSDam( ) ... 301
HrSSire( ), JrSSire( ), SrSSire( ) ... 301
HrSSSire( ), JrSSSire( ), SrSSSire( ) ... 301
HrStable( ), JrStable( ), SrStable( ) ... 302
HrStableAbbr( ), JrStableAbbr( ), SrStableAbbr( ) ... 302
HrStableID( ), JrStableID( ), SrStableID( ) ... 302
HrStableRetired( ), JrStableRetired( ), SrStableRetired( ) ... 303

**5 Track Work Information ... 303**

HrABTCount( ), JrABTCount( ), SrABTCount( ) ... 303
HrActivity( ), JrActivity( ), SrActivity( ) ... 304
HrTw ResumeDate( ), JrTw ResumeDate( ), SrTw ResumeDate( ) ... 306
HrBestGallopSect( ), JrBestGallopSect( ), SrBestGallopSect( ) ... 307
HrBTrial( ), JrBTrial( ), SrBTrail( ) ... 307
HrGallop1200Best( ), JrGallop1200Best( ), SrGallop1200Best( ) ... 309
HrGallop1200Stat( ), JrGallop1200Stat( ), SrGallop1200Stat( ) ... 309
HrGallop400Best( ), JrGallop400Best( ), SrGallop400Best( ) ... 310
HrGallop400Stat( ), JrGallop400Stat( ), SrGallop400Stat( ) ... 310
HrGallop800Best( ), JrGallop800Best( ), SrGallop800Best( ) ... 310
HrGallop800Stat( ), JrGallop800Stat( ), SrGallop800Stat( ) ... 311
HrGBTCount( ), JrGBTCount( ), SrGBTCount( ) ... 311
HrRGBTCount( ), JrRGBTCount( ), SrRGBTCount( ) ... 312
HrTw Count( ), JrTw Count( ), SrTw Count( ) ... 312

**6 Race and Meeting Information ... 314**

ComingDate( ) ... 316
CurRunHorses( ) ... 316
DateOfRace( ) ... 318
HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( ) ... 318
HrAccSectionalTimew ( ), JrAccSectionalTimew ( ), SrAccSectionalTimew ( ) ... 319
HrAveRating( ), JrAveRating( ), SrAveRating( ) ... 319
HrClassGroup( ), JrClassGroup( ), SrClassGroup( ) ... 319
HrCourse( ), JrCourse( ), SrCourse( ) ... 319
HrCourseDist( ), JrCourseDist( ), SrCourseDist( ) ... 320
HrCourseID( ), JrCourseID( ), SrCourseID( ) ... 320
HrCupName( ), JrCupName( ), SrCupName( ) ... 321
HrDate( ), JrDate( ), SrDate( ) ... 321
HrDayNight( ), JrDayNight( ), SrDayNight( ) ... 321
HrDayNote( ), JrDayNote( ), SrDayNote( ) ... 322
HrDaysAgo( ), JrDaysAgo( ), SrDaysAgo( ) ... 322
HrDistance( ), JrDistance( ), SrDistance( ) ... 322
HrDistTrack( ), JrDistTrack( ), SrDistTrack( ) ... 323
HrDvd( ), JrDvd( ), SrDvd( ) ... 323
HrDvdPlace( ), JrDvdPlace( ), SrDvdPlace( ) ... 324
HrDvdTrio( ), JrDvdTrio( ), SrDvdTrio( ) ... 325
HrDvdWin( ), JrDvdWin( ), SrDvdWin( ) ... 325
HrGoing( ), JrGoing( ), SrGoing( ) ... 325
HrGoingAbbr( ), JrGoingAbbr( ), SrGroingAbbr( ) ... 326
HrGoingID( ), JrGoingID( ), SrGoingID( ) ... 327
HrIsTurf( ), JrIsTurf( ), SrIsTurf( ) ... 327
HrLastSectionTime( ), JrLastSectionTime( ), SrLastSectionTime( ) ... 328
HrLastSectionTimew ( ), JrLastSectionTimew ( ), SrLastSectionTimew ( ) ... 329
HrMeetingNo( ), JrMeetingNo( ), SrMeetingNo( ) ... 329
HrPenetrometer( ), JrPenetrometer( ), SrPenetrometer( ) ... 329
HrRaceClass( ), JrRaceClass( ), SrRaceClass( ) ... 330
HrRaceID( ), JrRaceID( ), SrRaceID( ) ... 330
HrRaceNoInMeeting( ), JrRaceNoInMeeting( ), SrRaceNoInMeeting( ) ... 330

© Welkin Systems Limited

HrRaceNoInSeason( ), JrRaceNoInSeason( ), SrRaceNoInSeason( ) ... 331
HrRaceNote( ), JrRaceNote( ), SrRaceNote( ) ... 331
HrRacePrize( ), JrRacePrize( ), SrRacePrize( ) ... 331
HrRaceRating( ), JrRaceRating( ), SrRaceRating( ) ... 332
HrRaceRemark( ), JrRaceRemark( ), SrRaceRemark( ) ... 332
HrRacesInMeeting( ), JrRacesInMeeting( ), SrRacesInMeeting( ) ... 332
HrRaceTime( ), JrRaceTime( ), SrRaceTime( ) ... 332
HrRunners( ), JrRunners( ), SrRunners( ) ... 333
HrSeason( ), JrSeason( ), SrSeason( ) ... 333
HrSeason2( ), JrSeason2( ), SrSeason2( ) ... 334
HrSeasonRace( ), JrSeasonRace( ), SrSeasonRace( ) ... 334
HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( ) ... 335
HrSectionalTimew ( ), JrSectionalTimew ( ), SrSectionalTimew ( ) ... 335
HrSectionTime( ), JrSectionTime( ), SrSectionTime( ) ... 335
HrSectionTimew ( ), JrSectionTimew ( ), SrSectionTimew ( ) ... 336
HrPow erFigure( ), JrPow erFigure( ), SrPow erFigure( ) ... 336
HrPow erComments( ), JrPow erComments( ), SrPow erComments( ) ... 337
HrTrack( ), JrTrack( ), SrTrack( ) ... 337
HrTrackID( ), JrTrackID( ), SrTrackID( ) ... 337
HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( ) ... 338
HrTrackShift( ), JrTrackShift( ), SrTrackShift( ) ... 339
HrWinner( ), JrWinner( ), SrWinner( ) ... 339
HrWinnerID( ), JrWinnerID( ), SrWinnerID( ) ... 339
HrWinSeconds( ), JrWinSeconds( ), SrWinSeconds( ) ... 340
HrWinTime( ), JrWinTime( ), SrWinTime( ) ... 340
HrWinTimew ( ), JrWinTimew ( ), SrWinTimew ( ) ... 340
IrregularRace( ) ... 341

**7 Data Collection Assistant Functions ... 341**

AtStart( ) ... 341
Column( ) ... 342
CurRun( ) ... 343
GetColumn( ) ... 343
HrData( ), JrData( ), SrData( ) ... 344
HrHistoryList( ), JrHistoryList( ), SrHistoryList( ) ... 345
IfAtTop( ) ... 346
IfCurRun( ) ... 347
RaceVar( ) ... 348
StartVar( ) ... 351

**8 Statistics on Past Records ... 353**

HrAverage( ), JrAverage( ), SrAverage( ) ... 354
HrAverageBy( ), JrAverageBy( ), SrAverageBy( ) ... 355
HrAverageNZ( ), JrAverageNZ( ), SrAverageNZ( ) ... 357
HrCount( ), JrCount( ), SrCount( ) ... 358
HrCountBy( ), JrCountBy( ), SrCountBy( ) ... 360
HrCountNZ( ), JrCountNZ( ), SrCountNZ( ) ... 366
HrMaximum( ), JrMaximum( ), SrMaxmum( ) ... 367
HrMaxStartID( ), JrMaxStartID( ), SrMaxStartID( ) ... 369
HrMinimum( ), JrMinimum( ), SrMinimum( ) ... 370
HrMinStartID( ), JrMinStartID( ), SrMinStartID( ) ... 371
HrStatList( ), JrStatList( ), SrStatList( ) ... 373
HrSum( ), JrSum( ), SrSum( ) ... 374
HrSumBy( ), JrSumBy( ), SrSumBy( ) ... 375
HrSumCount( ), JrSumCount( ), SrSumCount( ) ... 377
HrXAverage( ), JrXAverage( ), SrXAverage( ) ... 379
HrXMaximum( ), JrXMaximum( ), SrXMaximum( ) ... 380
HrXMaxStartID( ), JrXMaxStartID( ), SrXMaxStartID( ) ... 381
HrXMinimum( ), JrXMinimum( ), SrXMinimum( ) ... 382
HrXMinStartID( ), JrXMinStartID( ), SrXMinStartID( ) ... 383

HrXSum( ), JrXSum( ), SrXSum( ) ... 385
StdTime( ) ... 386
StdTimew ( ) ... 387
9 Statistics in a Race ... 387
RunnersAvg( ) ... 387
RunnersCnt( ) ... 388
RunnersFind( ) ... 389
RunnersList( ) ... 390
RunnersMax( ) ... 390
RunnersMin( ) ... 391
RunnersSum( ) ... 392
10 Arithmetic Functions ... 393
Average( ) ... 393
AverageNZ( ) ... 393
Int( ) ... 394
Maximum( ) ... 394
Minimum( ) ... 395
Over( ) ... 395
Sum( ) ... 396
11 String Functions ... 396
At( ) ... 397
Atc( ) ... 397
Left( ) ... 397
Like( ) ... 398
LTrim( ) ... 398
Occurs( ) ... 399
Padr( ) ... 400
Replicate( ) ... 400
Right( ) ... 401
StrExtract( ) ... 401
SubStr( ) ... 402
Trim( ) ... 403
12 List Functions ... 404
ListAscItem( ) ... 404
ListAscVal( ) ... 407
ListAverage( ) ... 407
ListCol( ) ... 408
ListColVal( ) ... 408
ListCommon( ) ... 408
ListCount( ) ... 408
ListDescItem( ) ... 409
ListDescVal( ) ... 410
ListFilter( ) ... 410
ListItem( ) ... 411
ListSort( ) ... 412
ListSortDesc( ) ... 414
ListsortDescVal( ) ... 414
ListSortVal( ) ... 414
ListStdDev( ) ... 415
ListSum( ) ... 415
ListVal( ) ... 416
RListItem( ) ... 417
RListVal( ) ... 417
13 Conversion Functions ... 418
Cnv( ) ... 418
Conv( ) ... 420

© Welkin Systems Limited

DateStr( ) ... 420
DToC( ) ... 421
Evl( ) ... 421
GCnv( ) ... 422
GConv( ) ... 423
QStr( ) ... 424
NVL( ) ... 425
SCnv( ) ... 425
SConv( ) ... 426
SToT( ) ... 427
Str( ) ... 427
StrDiff( ) ... 428
TblAddVal( ) ... 428
TblWrite( ) ... 429
Transform( ) ... 430
TToS( ) ... 430
Val( ) ... 431

**14 Multimedia & Internet Functions ... 431**

HrBTVideo( ), JrBTVideo( ), SrBTVideo( ) ... 431
HrPostRace( ), JrPostRace( ), SrPostRace( ) ... 432
HrPreRace( ), JrPreRace( ), SrPreRace( ) ... 433
HrPosPhoto( ), JrPosPhoto( ), SrPosPhoto( ) ... 434
HrRaceReplay( ), JrRaceReplay( ), SrRaceReplay( ) ... 435
HrRacingFocus( ), JrRacingFocus( ), SrRacingFocus( ) ... 436

**15 Miscellaneous Functions ... 437**

Box( ) ... 437
Compare( ) ... 439
Date( ) ... 440
Edition( ) ... 440
Empty( ) ... 441
Gel( ) ... 441
ICase( ) ... 442
Iif( ) ... 442
InList( ) ... 443
IsNull( ) ... 443
Join( ) ... 443
Lang( ) ... 444
Match( ) ... 444
Month( ) ... 445
Same( ) ... 446
Season( ) ... 446
Year( ) ... 447

**16 VFP Functions & Operators ... 447**

VFP Functions ... 447
VFP Operators ... 448

## Part XVII How To 450

**1 Find horses of this race met in past starts ... 450**

**2 Find horses which win in 60 days after a stable change ... 452**

**3 See performance enhanced by a stable change ... 457**

## Part XVIII Questions & Answers 464

**1 Questions You May Asked After Upgrading to 9.0 ... 464**

**2 How Can I Display Chinese in English Windows ... 465**

© Welkin Systems Limited

3 Chinese Characters Becomes Garbage ..... 466
4 RaceMate Frozen When Starter View Being Opened ..... 467

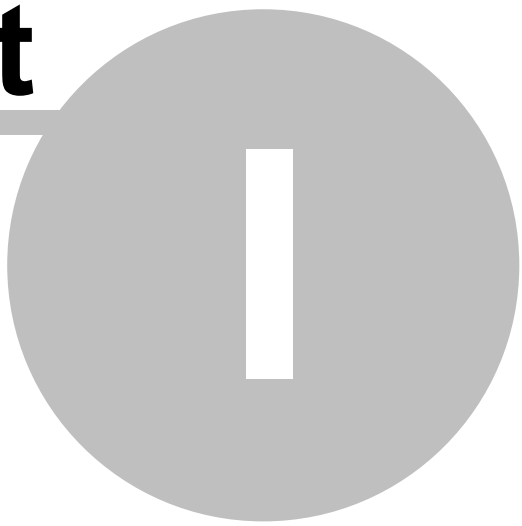
Part XIX Technical Reference 470
1 Data Range for Hong Kong Edition ..... 470

Part XX License Agreement 474

Index 477

© Welkin Systems Limited

# Part



# 1 Introduction to RaceMate

Thank you for using RaceMate, the most popular and powerful software on racing analyses.

This documentation is designed to be an easy-to-use aid for RaceMate. It is organized to take you through basic usage to advanced troubleshooting. Besides using this Help file, you can get more help by downloading RaceMate manuals from our web site http://www.RaceMateNet.Com. You may also see a lot of techniques and examples about RaceMate usage at http://RaceMate.wordpress.com.

You are welcome to send your questions and feedback to Contact@RaceMateNet.Com.

**>>> See the following topics in this section to learn more about RaceMate <<<**

- Main Features
- System Requirements
- Installing RaceMate
- Registering and Activating RaceMate
- Installing into More Than One Computer
- About Upgrading Windows
- Remarks about Inputting Chinese

## 1.1 Main Features

RaceMate is a powerful race data analysis tool. Its huge database provides you with all necessary data for your betting analysis. Its versatile features help you effectively find out investment hints and improve chances of winning. You will be much impressed by the lightning speed of RaceMate because its data is driven by the World's fastest desktop database engine Microsoft Visual FoxPro. Since its database resides in your local hard disk, there is no need to dedicate an Internet connection while you are using the software. You may install RaceMate in a notebook and use it anytime and anywhere.

Here are the some of the main features of RaceMate:

**Huge Database**

- Data are always up-to-date. (Simply carry out a 1-minute data download to keep your database up-to-update.)
- Comprehensive data since 1987, including races, track work, barrier trials, swimming, entries, forms, results, trainers, jockeys, records of horses, pedigree information (includes 3-generation bloodline charts and listing of horses of the same dam, sire or sire of dam), and much more.
- Past records of horses, jockeys and trainers can dynamically be displayed in forms, starter views, and race results. You may choose the number of history lines, or even display only the records satisfying a set of criteria that you specified.
- Multiple track work views for effectively identifying training patterns.
- Talented data sorting and grouping by 1 or 2 mouse clicks.

**Multimedia Resources**

© Welkin Systems Limited

- Automatic hyper-linking to multimedia resources of the Hong Kong Jockey Club, including photos of running, videos of track work, race replay and pre-race studies.
- Multimedia resources are integrated in records of races, horses, jockeys, and stables. The resource are always ready for accessing without searching.
- Integrating your personal multimedia resources with race and horse records.

**Excel-Like Spreadsheet Features**

- RaceMate provides Excel-Like spreadsheet features. You may set formulae to do powerful calculation among columns.
- There are hundreds of functions for calculations and statistics.
- On the spreadsheet you may retrieve past and future records or do statistics of horses, jockeys and stables.
- You may import data from external files into your spreadsheet.

**Computer Rating and Data Analysis Models**

- RaceMate's Flemming Model and Young's Betting Index provide you with accurate computer rating about starters.
- You are encouraged to build your own analysis models by using RaceMate's Model Construction Language (MCL).
- Validating your model by fitting it with real past records. Improving your models based on validation results.

**Powerful Statistics**

- You are free to set your own statistical criteria, which are limited only by your imagination and availability of data.
- Criteria setting interface is kept simple evening for complicated statistics requirements.
- RaceMate's Model Construction Language can readily be used in statistics. Professional statistics that are typically done by computer programmers can now be carried out by laypersons.
- You may create data conversion tables for converting text data to numbers. Conversion tables are designed for statistics and analysis models.
- Open related forms by mouse-clicking on statistics results.
- Your personal notes and ratings can be part of statistical criteria.

**Flexible Data Searching**

- Data Searching functions use the same engine of RaceMate's statistics system, and inherit all the capability and flexibility of the features of the statistics system. Data search can be carried out in almost any data fields.
- You may put the results of a search in a list and open the races by a mouse click.
- Incremental search can be carried out in all data tables.
- Provides different Chinese character indexing systems, e.g., Cantonese phonetic index. Chinese name searching is made simple.

**Electronic Personal Notes and Rating System**

- Add your notes and ratings directly in records of horses, jockeys, stables, races, track work.
- Notes can dynamically be displayed with past records of horses in forms, starter views, and race results.
- No limit on sizes of the notes.
- Personal notes can be search and are fully integrated with the statistics and data searching system of RaceMate.

© Welkin Systems Limited

**Custom Screen layouts and User-Defined Reports**

- You may the layouts of starters views by adding new data columns or deleting existing columns.
- Statistical figures can be inserted into user-defined columns.
- You may insert unlimited columns of personal notes and rating into a starters view.
- Using RaceMate's User Report Designer, you may design your our reports for printing data of races.

Remarks: Track work data of RaceMate Hong Kong Editions starts from 1992. Track work data for other editions may be simpler and start later.

## 1.2 System Requirements

RaceMate runs on an IBM PC-compatible computer. The minimum system requirements of RaceMate are as follows:

**Hardware Requirements**

- IBM PC-compatible CPU
- 32 MB RAM (128MB RAM recommended)
- Internet Connect or Modem (for updating your RaceMate data)
- At least 250Mb free hard disk space (for future data growth and working space for user-define views, 350Mb are recommended)

**Operating System**

RaceMate runs on any of the following operating systems:

- Windows 98
- Windows Me
- Windows 2000
- Windows XP
- Windows Vista
- Windows 7 or above

Note: RaceMate does not run on Windows 3.x and Windows NT

**Windows Security Rights**

- RaceMate may update its program files while normal data update. Therefore it must be installed into a folder with "modify" right. The folder "Program Files" of Windows Vista or above have been made read-only. Therefore please don't install RaceMate into this folder. "C:\RaceMate" is the default installation folder for RaceMate.
- RaceMate will write a couple of static files into the folders c:\Windows and c:\Program Files\Common Files.

**Displaying Chinese Characters**

If you want to display Chinese data of RaceMate, you are required to have one of the following systems:

- Chinese Windows (Traditional Chinese or Simplified Chinese)
- English Windows with RichWin or other Chinese enabling tools.

**Inputting Chinese Characters**

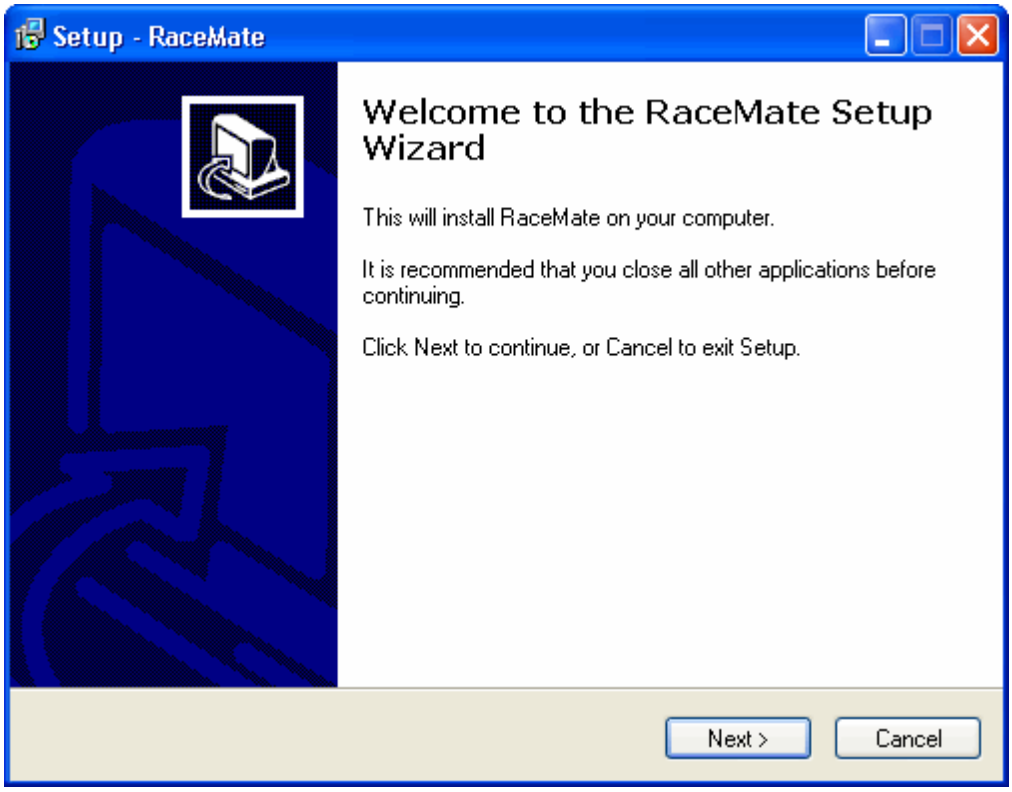
© Welkin Systems Limited

You may use any input method, including pen input and voice recognition, supported by your Windows to input Chinese characters.

## 1.3 Installing RaceMate

### 1. Inserting Installation CD

Insert your RaceMate installation CD into your CD ROM drive. The setup program of RaceMate will be run automatically. If the Autorun feature of your CD ROM drive does not work, please double-click on the setup file in the CD

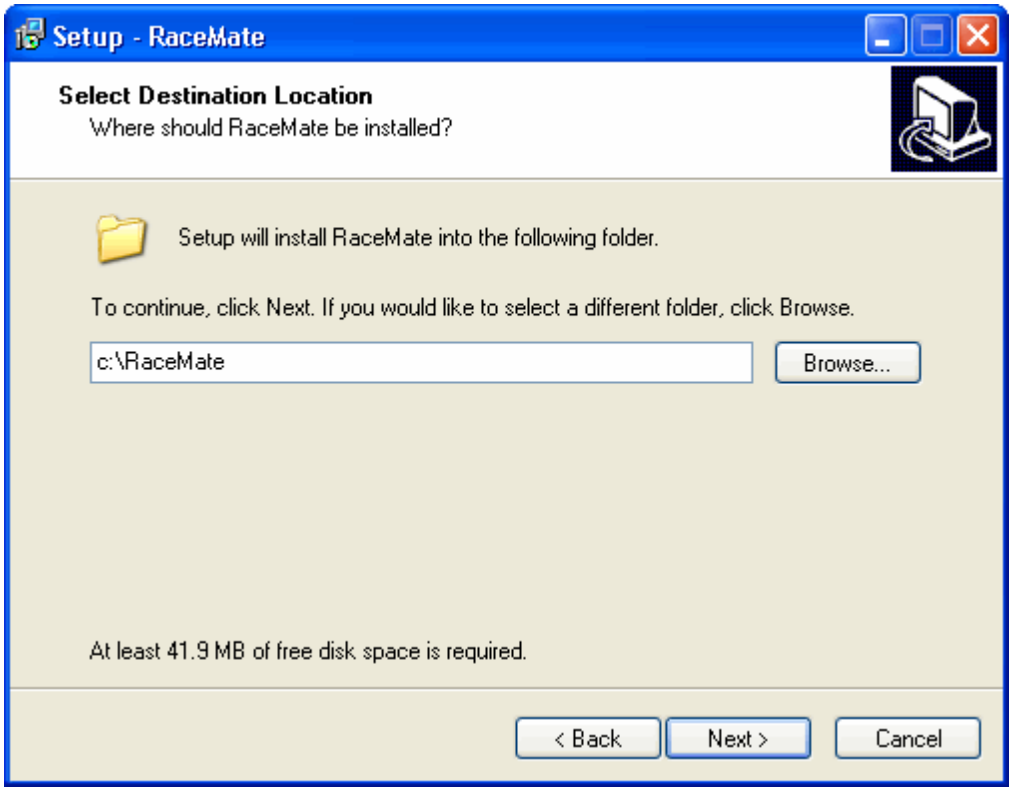

Setup - RaceMate
Welcome to the RaceMate Setup Wizard
This will install RaceMate on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

Click on the [Next] button. The Read Me File dialogue will appear. The dialog shows important information about the RaceMate software including the software license agreement. Please scroll up and down to read the message carefully in the message box. If the License Agreement is okay for you, please choose [I accept the agreement] and click on [Next].

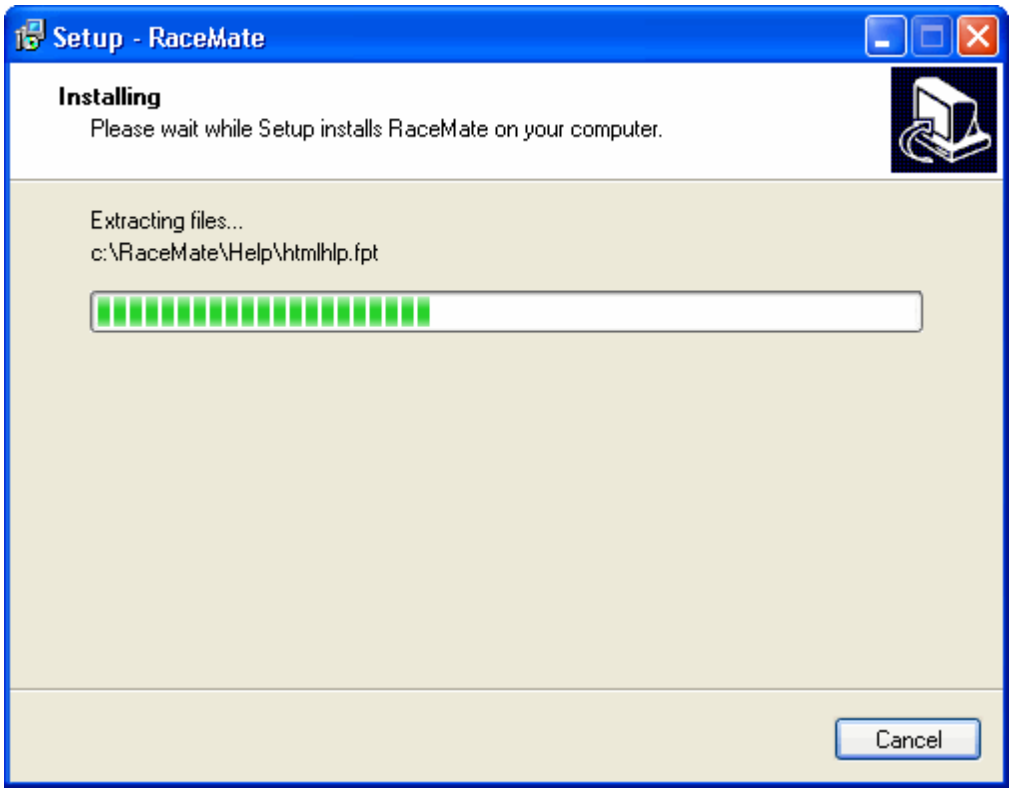
### 2. Choosing Software Folder

The following folder selection screen will appear. Normally, we will accept the folder suggested by the Setup Program. If you want to install RaceMate into another folder, click on [Browse] to choose a folder. RaceMate may update its program files while normal data update. Therefore it must be installed into a folder with "modify" right. The folder "Program Files" of Windows Vista or above have been made read-only. Therefore please don't install RaceMate into this folder.

© Welkin Systems Limited


Setup - RaceMate
Select Destination Location
Where should RaceMate be installed?
Setup will install RaceMate into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
c:\RaceMate
Browse...
At least 41.9 MB of free disk space is required.
< Back
Next >
Cancel

After your have clicked on the [Next] button, the Setup Program will confirm with you the destination location. If the path is correct, click on the [Install] button. You will see the following screen.

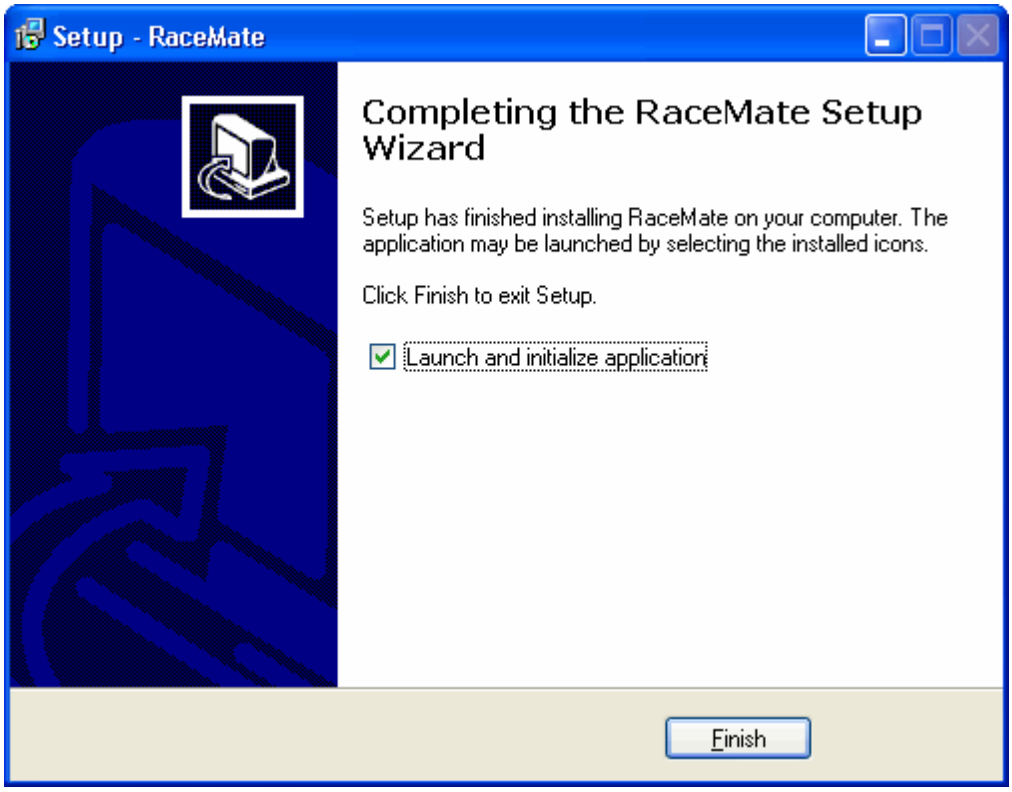



© Welkin Systems Limited

**Installation Completes**

After the file transferring process completes, you will see the dialogue box as shown in the following figure. Please push the [Finish] button to start RaceMate. After RaceMate is started, please follow the steps in next section to register you software and create the database of RaceMate.

If there existed some system files that were out-of-date and updated by the setup program, you will be prompted to restart your computer and the above screen will not be shown. After your computer is restarted, please click on the RaceMate icon on you desktop, which was created by the setup program, to start RaceMate.


Setup - RaceMate
Completing the RaceMate Setup Wizard
Setup has finished installing RaceMate on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch and initialize application
Finish

## 1.4 Registering and Activating RaceMate

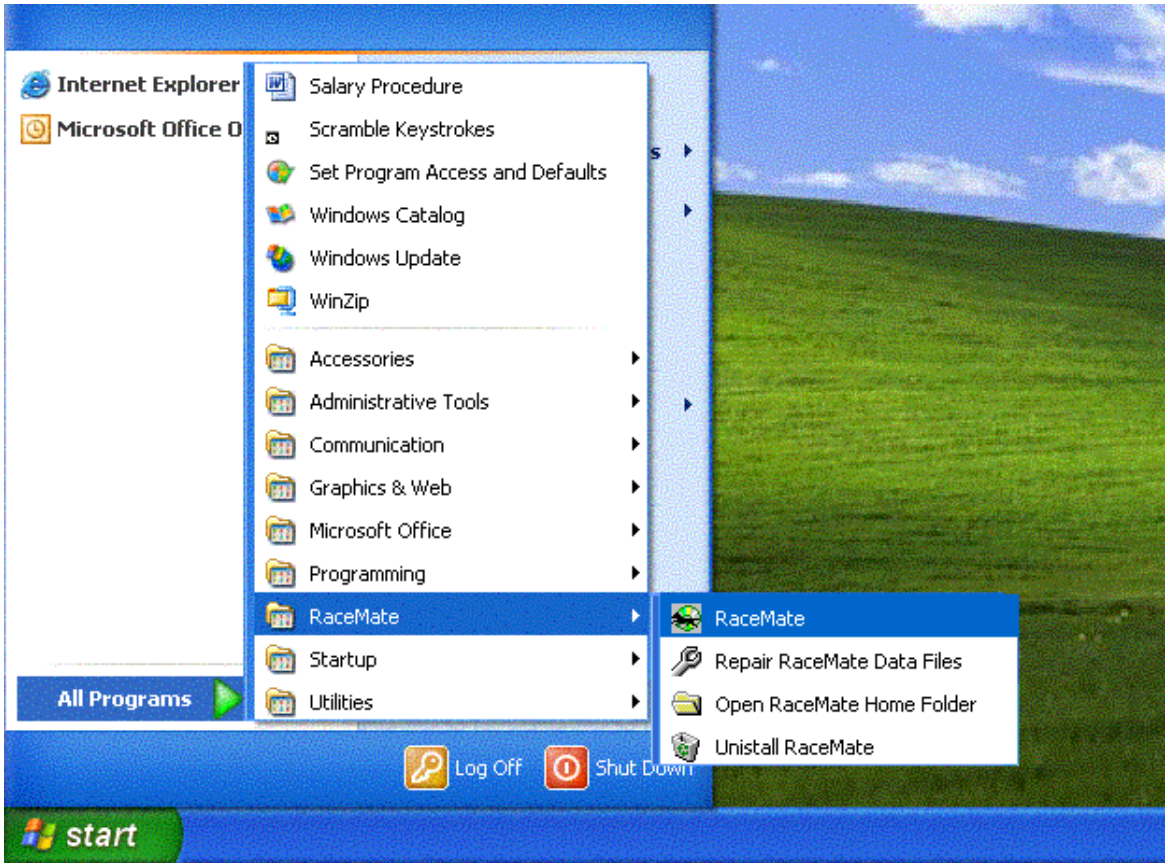
Note: Content of this page is only applicable to RaceMate packages purchased after March 2004. Packages purchased before March 2004 uses an external protection key lock. The following registration and activation procedure is not applicable.

After RaceMate installation is completed, you must follow the steps below to register and activate your software.

**Start RaceMate**

Click on the Start button of Windows and select the RaceMate program group and then click on RaceMate to run the program.

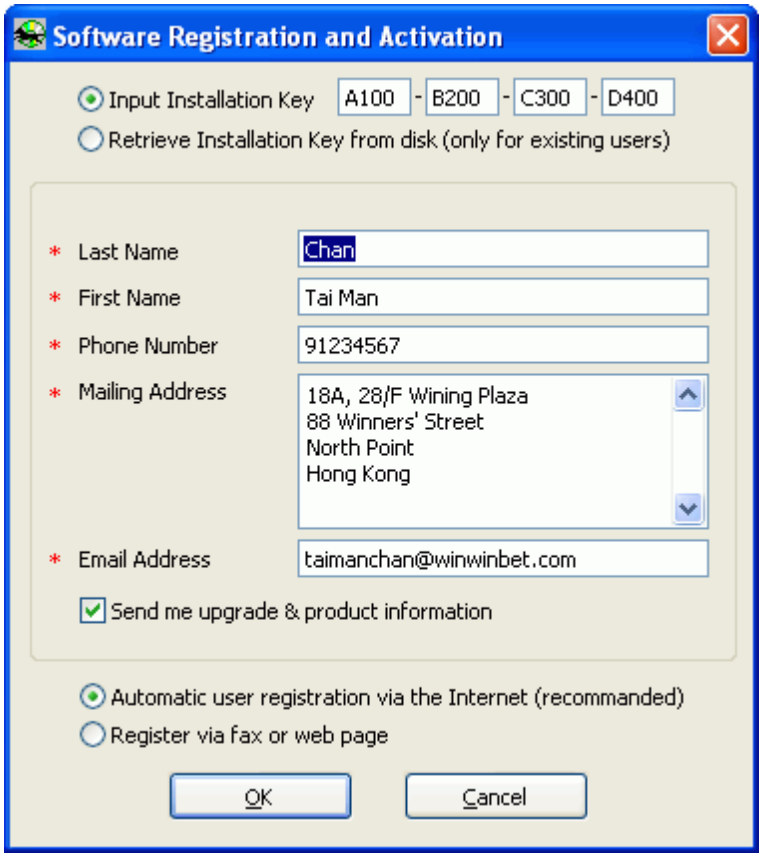
© Welkin Systems Limited



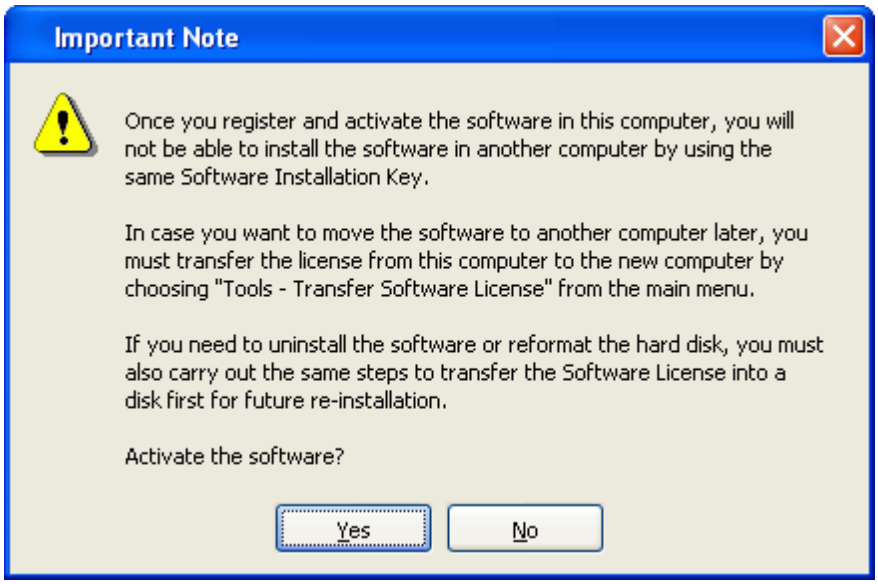

### User Registration

The first time you start RaceMate, the following registration form will be shown. Please enter the Software Installation Key into the boxes at the top of the form and other information into the edit boxes below. If your computer is able to be connected to the Internet, please establish the connection and select "Automatic user registration via the Internet". Otherwise please select "Register via fax or web page".

© Welkin Systems Limited



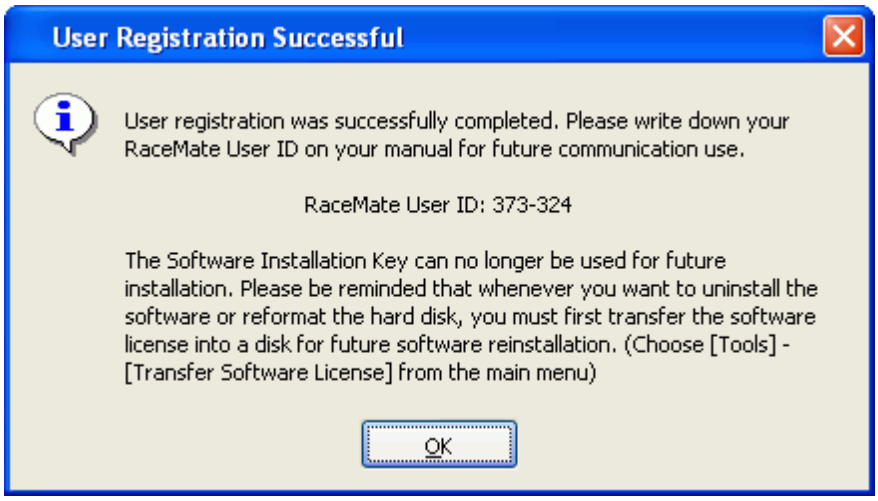

You will then see the following screen. Please read carefully the message on the screen and click on [Yes].


Important Note
Once you register and activate the software in this computer, you will not be able to install the software in another computer by using the same Software Installation Key.
In case you want to move the software to another computer later, you must transfer the license from this computer to the new computer by choosing "Tools - Transfer Software License" from the main menu.
If you need to uninstall the software or reformat the hard disk, you must also carry out the same steps to transfer the Software License into a disk first for future re-installation.
Activate the software?
Yes
No

**Automatic User Registration via the Internet**

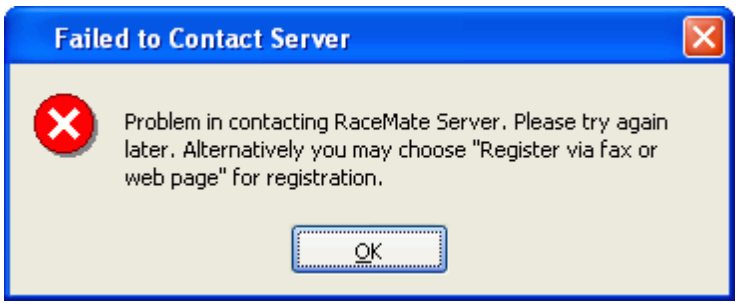
If you have chosen "Automatic user registration via the Internet" and successfully activated your software, you will see the following message:

© Welkin Systems Limited


User Registration Successful
User registration was successfully completed. Please write down your RaceMate User ID on your manual for future communication use.
RaceMate User ID: 373-324
The Software Installation Key can no longer be used for future installation. Please be reminded that whenever you want to uninstall the software or reformat the hard disk, you must first transfer the software license into a disk for future software reinstallation. (Choose [Tools] - [Transfer Software License] from the main menu)
OK

Sometimes computer firewalls, if any, may block the communication of RaceMate with its data server. See the topic FirewallBlocking for more information.

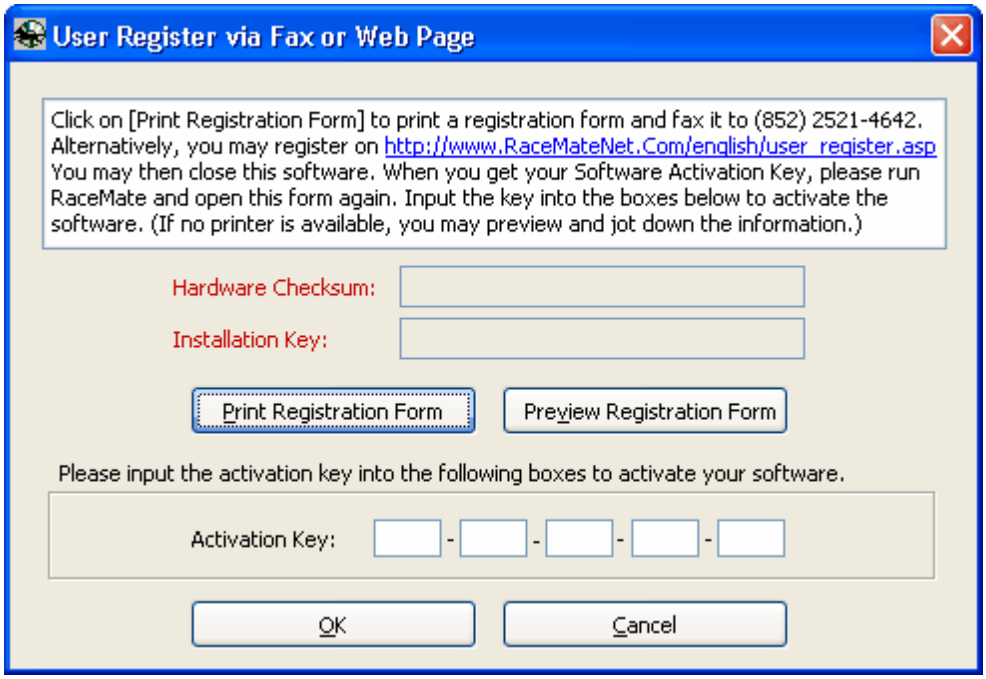
In case the RaceMate registration server is too busy and can't be connected or RaceMate is blocked by the firewall of your computer, you will see the following message. You may click on OK and retry the steps later. Alternatively, you may change your registration method to "Register via fax or web page"


Failed to Contact Server
Problem in contacting RaceMate Server. Please try again later. Alternatively you may choose "Register via fax or web page" for registration.
OK

**Register via fax or web page**

If you have chosen "Register via fax or web page", the "Important Notes" dialog (see a screen capture in the section above) will be shown. After you have read carefully the message on it and clicked on the [OK] button, you will see the form below. Click on the [Print Registration Form] button to print the registration form and fax it to (852) 2521 4642 or visit the webpage http://www.RaceMateNet.com/english/User_register.asp on a computer which is connecting to the Internet. (If no printer is available, you may preview and jot down the information on a paper.)

© Welkin Systems Limited

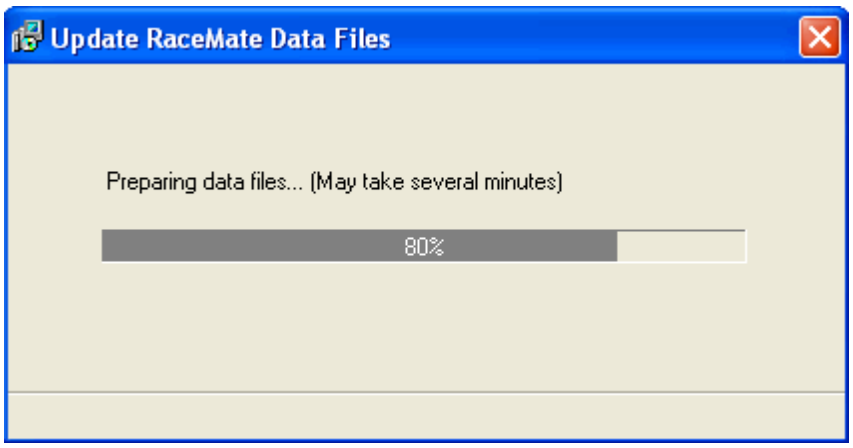

User Register via Fax or Web Page
Click on [Print Registration Form] to print a registration form and fax it to (852) 2521-4642.
Alternatively, you may register on http://www.RaceMateNet.Com/english/user_register.asp
You may then close this software. When you get your Software Activation Key, please run RaceMate and open this form again. Input the key into the boxes below to activate the software. (If no printer is available, you may preview and jot down the information.)
Hardware Checksum:
Installation Key:
Print Registration Form
Preview Registration Form
Please input the activation key into the following boxes to activate your software.
Activation Key:
OK
Cancel

Once we get the information, we will send you a Software Activation Key. Before you got your Software Activation Key, you may close RaceMate. Once you get the key, please open the above dialog again and input the key into the boxes below at the lower part and click [OK]. If you see the "User Registration Successful" message box as show in previous section, your RaceMate is ready for use.

Important Notes: Software Activation Key is hardware dependent and can only be used with the computer that you carry out the registration procedure. Once you get the Activation Key you must proceed to activating the software. Even if now you want to install RaceMate into another computer, you must first activate the software in this computer and then carry out a Moving or Uninstalling RaceMate to move the software to the computer that you want.

### Creating New Database

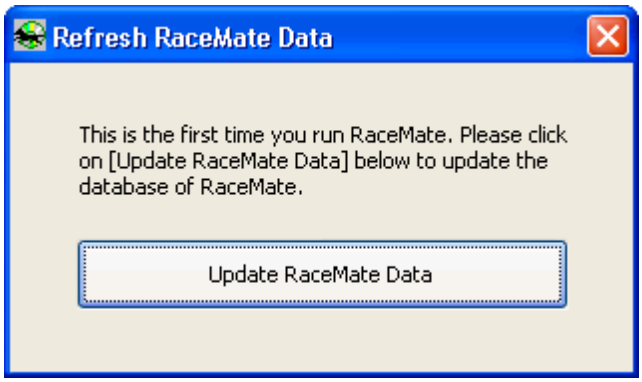
After the software is activated, RaceMate will begin to create a new database. When it is working you will see a screen similar to the following. Depending on the speed of your computer, the process will take from several minutes to over ten minutes.




© Welkin Systems Limited

### First Time Starting RaceMate

When database is created, you will see the following form:


Refresh RaceMate Data
This is the first time you run RaceMate. Please click on [Update RaceMate Data] below to update the database of RaceMate.
Update RaceMate Data

Please click on [Download Setting] to set up your data update method (see Data Update if you have any problem), and the click on the [Update RaceMate Data] button to refresh your RaceMate data.

## 1.5 Installing into More Than One Computer

You can install RaceMate into any number of computers. You can view data in any of them. However only one of them can download new data. We called the RaceMate which can download new data an "active RaceMate", and the others "inactive RaceMate". If you want to make an inactive RaceMate active, you have to transfer the "date update right" (license) from the active RaceMate into it.

**Remark:**
Content of this page is only applicable to RaceMate packages purchased after March 2004. Packages purchased before March 2004 uses a hardware protection key lock. The following license transfer procedure is not applicable.

### Installing Your First Copy of RaceMate

Follow the steps on the page Installing RaceMate to install your first copy of RaceMate.

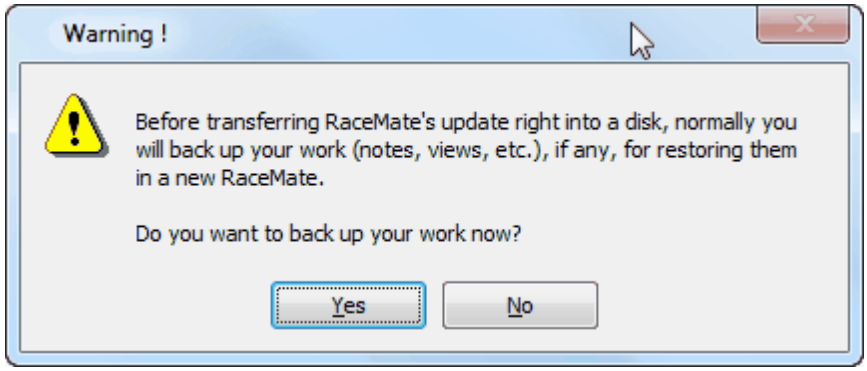
### Preparing License File for 2nd Copy: Exporting Data Update Right into Movable Disk

Follow the steps below to transfer the Update Right (License) of your RaceMate from your computer into a disk file.

**Remark:**
If you are a Dropbox (or other similar tools) user, you may save the exported file into a Dropbox folder. Then you can read the license file from other computers without any disk. Dropbox is free. You can learn more about the tool and download it on the page http://www.dropbox.com.

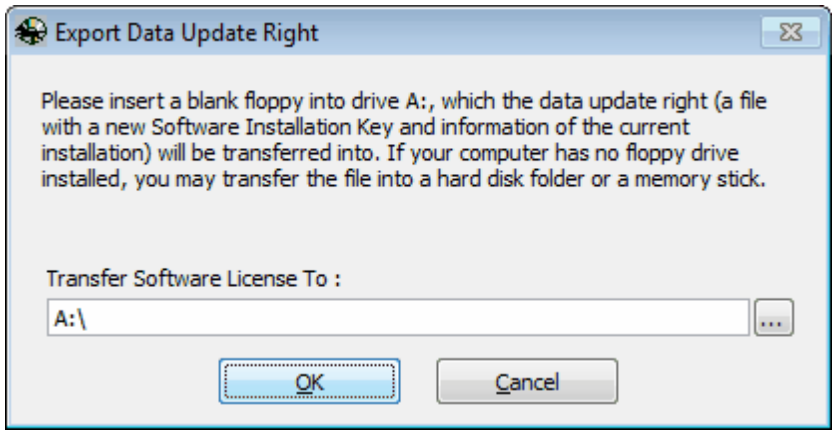
1. Choose [Export Data Update Right] (or Transfer Software License, previously) from the [Tools] menu. You will see the following form asking you to back up your work. Normally you should click on [Yes] to back up all your work into a disk.

© Welkin Systems Limited

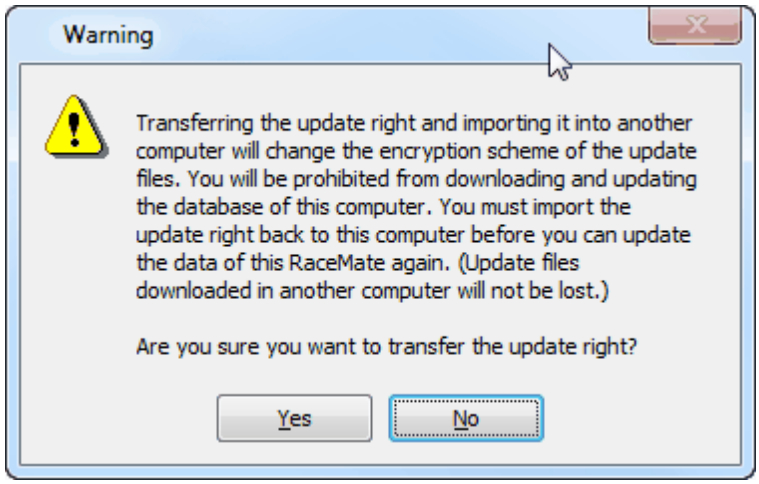

Warning !
Before transferring RaceMate's update right into a disk, normally you will back up your work (notes, views, etc.), if any, for restoring them in a new RaceMate.
Do you want to back up your work now?
Yes
No

After clicking on [Yes], follow the steps in the page Backing Up and Restoring User Data to back up your work. If [No] is chosen, you will see the form in the following step.

2. The form "Export Data Update Right" will be shown as follows.

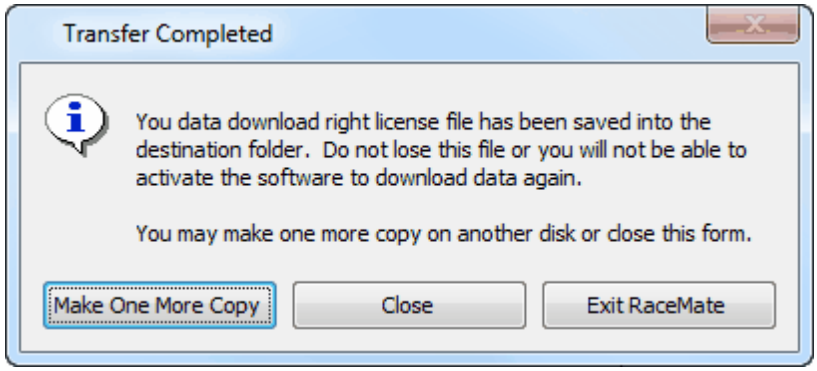

Export Data Update Right
Please insert a blank floppy into drive A:, which the data update right (a file with a new Software Installation Key and information of the current installation) will be transferred into. If your computer has no floppy drive installed, you may transfer the file into a hard disk folder or a memory stick.
Transfer Software License To :
A:\
...
OK
Cancel

3. Insert a blank floppy disk into drive A: and click on [OK]. If you are using a USB drive or a Dropbox folder, you must change "A:\" to the proper folder name.The system will show the following warning message. After transferring the license of your RaceMate into a disk, you will still be able to start RaceMate in your computer. However you will not be able to update its data.


Warning
Transferring the update right and importing it into another computer will change the encryption scheme of the update files. You will be prohibited from downloading and updating the database of this computer. You must import the update right back to this computer before you can update the data of this RaceMate again. (Update files downloaded in another computer will not be lost.)
Are you sure you want to transfer the update right?
Yes
No

4. If the data update right is successfully transferred into the disk, you will see the following message. If you want to make anther copy as a back-up, you may push the [Make One More Copy] button.

© Welkin Systems Limited


Transfer Completed
You data download right license file has been saved into the destination folder. Do not lose this file or you will not be able to activate the software to download data again.
You may make one more copy on another disk or close this form.
Make One More Copy
Close
Exit RaceMate

5. When you click on [Close], RaceMate will continue to run with the data update feature will be disabled. To enabled it again, you have to carry out the "import the data update right" procedure.

**On the Other Computer: Importing Data Update Right from the Disk to a New Computer**

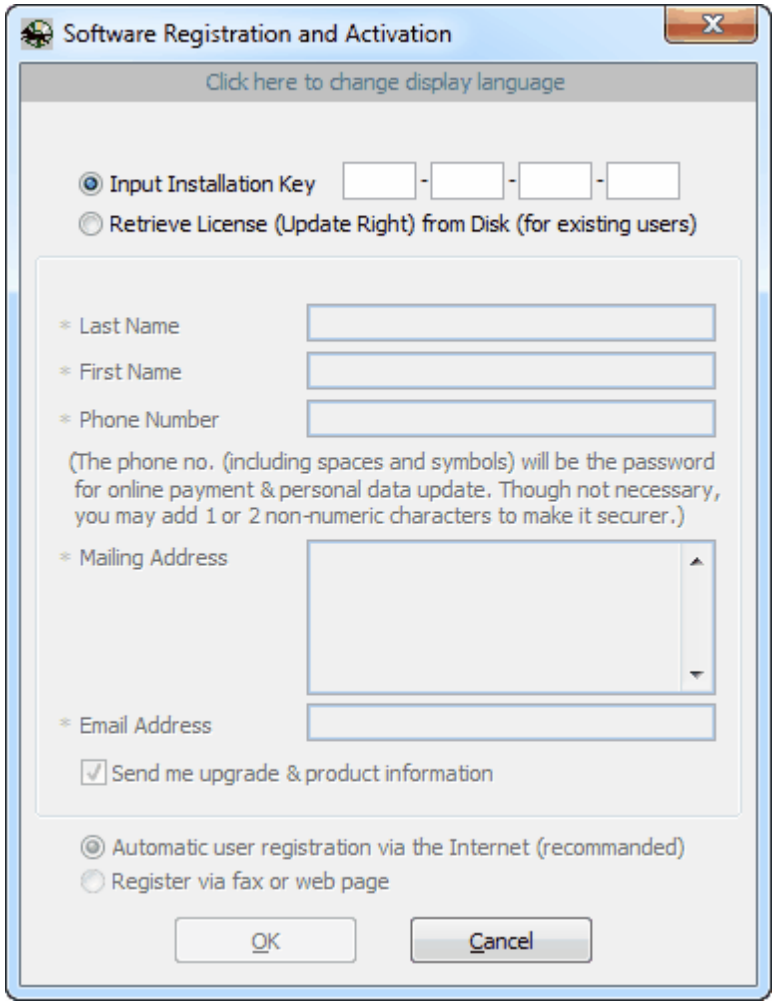
After you have completed the "Export Data Update Right" procedure above, please follow the procedure below to transfer the right into a new computer.

6. If there is already a RaceMate in this computer, please choose [Import Data Update Right] from the [Tools] menu of RaceMate. If there is no RaceMate, you need to install the software. Please download a latest setup program from one of the following links.

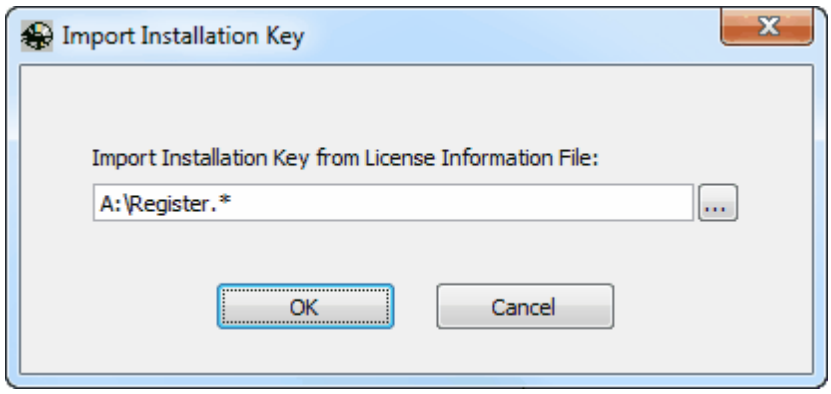
   Hong Kong edition: http://www.racematenet.com/setup/hk/setup-hk.exe
   Macau edition: http://www.racematenet.com/setup/macau/setup-macau.exe

7. After choosing [Import Data Update Right] (or after RaceMate is installed), you will see the following Software Registration Form.

© Welkin Systems Limited


Software Registration and Activation
Click here to change display language
Input Installation Key
Retrieve License (Update Right) from Disk (for existing users)
Last Name
First Name
Phone Number
(The phone no. (including spaces and symbols) will be the password for online payment & personal data update. Though not necessary, you may add 1 or 2 non-numeric characters to make it securer.)
Mailing Address
Email Address
Send me upgrade & product information
Automatic user registration via the Internet (recommanded)
Register via fax or web page
OK
Cancel

8. Choose [Retrieve License from Disk]. You will see the following form.




9. If you have saved your license file in another folder other than A:, change A: to the folder name. Otherwise insert disk A:. Then push [OK].
10. If the disk is successfully read, RaceMate will continue to run.
11. If you have backed up your personal data in step 1, pull down the [Tools] menu and choose [Restore User Data Input and Changes from Previous Backup] to restore them.

**Important:**
- After a license file has successfully been imported into an inactive RaceMate, it becomes useless and can't be imported into other RaceMates anymore.
- If you want to change a new hard disk or update the data of an inactive RaceMate, you must

© Welkin Systems Limited

export a new license file from the active RaceMate.

### Move Update Right Between Computers

You can install RaceMate into as many computers as you like and move the update right among all copies whenever necessary.

## 1.6 About Upgrading Windows

### Upgrading to Windows XP or Earlier Versions

New Windows on Another Computer

If you are upgrading your Windows from Windows 98 to Windows XP, please follow the steps in the section Moving or Uninstalling RaceMate.

In-Place Upgrade

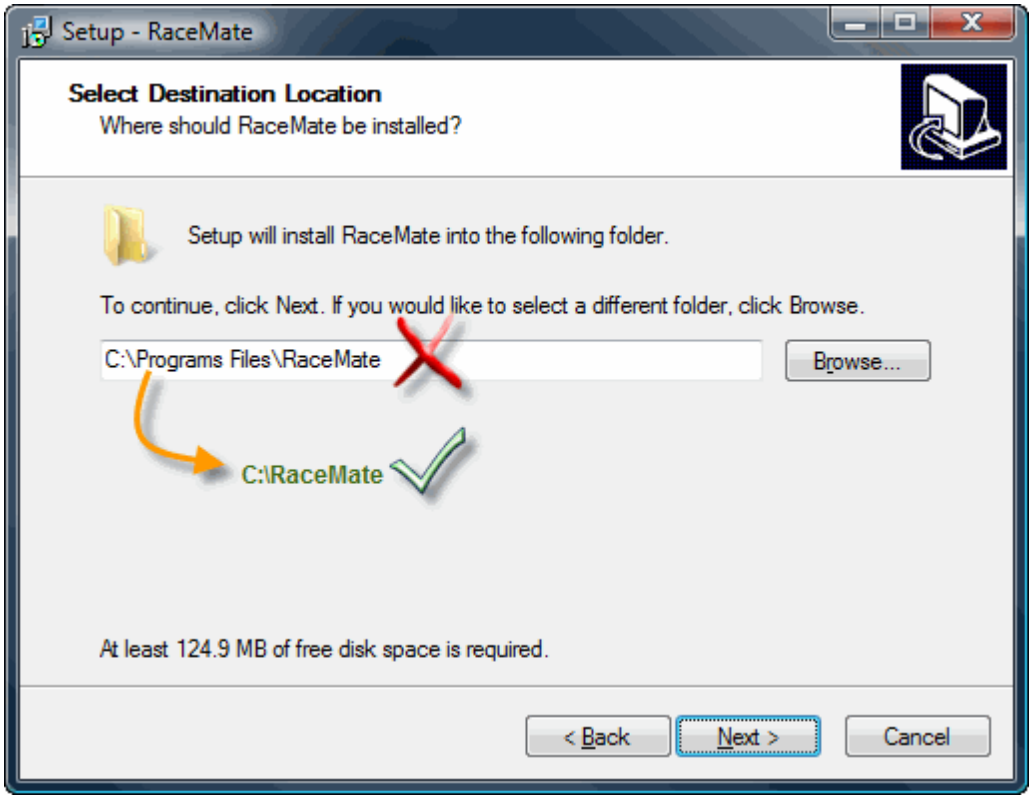
If you are going to upgrade an old version Windows to Windows XP (or earlier) in your existing computer, no special steps about RaceMate are needed. You may simply carry out the normal procedure for upgrading Windows.

### Upgrading to Windows Vista, Windows 7, or Above

No matter you are doing an in-place upgrade or moving RaceMate into a new Vista or Windows 7 (or above) computer, you need to follow the steps for moving RaceMate in the section Moving or Uninstalling RaceMate.

If you are doing an in-place upgrade, the setup program will use your existing RaceMate folder as destination. If your existing RaceMate is in C:\Program Files\RaceMate, don't forget to change the destination to C:\RaceMate.

© Welkin Systems Limited


Setup - RaceMate
Select Destination Location
Where should RaceMate be installed?
Setup will install RaceMate into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Programs Files\RaceMate
Browse...
C:\RaceMate
At least 124.9 MB of free disk space is required.
< Back
Next >
Cancel

## 1.7 Remarks about Inputting Chinese

RaceMate can be used with both Traditional Chinese Windows and Simplified Chinese Windows. Please note the following remarks:

1. Traditional Chinese Windows and Simplified Chinese Windows are using different internal coding systems. Never restore your Chinese notes that you created in a Simplified Chinese Windows into a Traditional Chinese Windows, or vice versa. Otherwise the Chinese characters will become garbage.
2. The race data and system messages of RaceMate will always be displayed in Traditional Chinese characters no matter you run it on a Traditional Chinese Windows or a Simplified Chinese Windows. However, Simplified Chinese characters that you input in RaceMate will not be converted into Traditional Chinese characters and will always be displayed as Simplified Chinese characters.

© Welkin Systems Limited

# Part II

# 2 Release Notes

This chapter will state the new features and changes in each release of RaceMate.

**>>> Subtopics of this section <<<**


- What is New in RaceMate 11.5
- What is New in RaceMate 10.5
- What is New in RaceMate 10.0
- What is New in RaceMate 9.8
- What is New in RaceMate 9.5
- What is New in RaceMate 9.0
- What is New in RaceMate 8.0
- What is New in RaceMate 7.0


## 2.1 What is New in RaceMate 11.5

RaceMate has been upgraded to version 11.5 in October 2011. Below are the changes since version 10.5.

**MCL Variables**

You can define up to 9 MCL variables in an MCL function of a single view. MCL variables of one view are independent of those of other views. MCL variables are quoted by brace brackets. A default value can be putting the value at the right and a ":" between the variable number and the value. For example:

{1} define MCL variable number 1

{1:28} define MCL variable 1 and set its default value to 28

- You can use the new [Set/Choose MCL Variables...] button to quickly change the set statistics range by choosing new values for MCL variables.
- When you double-click on names of horses, jockeys and stables on a starter view, corresponding forms will open to show the information of the horse, jockey and stable that you click on. You can pass as arguments the values of the MCL variables defined in the starter view to the chained form.

Please learn how to use MCL variabels from the following pages.
Introduction to MCL Variables.
Setting Values & Choosing MCL Variables
Using MCL Variables As View Arguments
MCL Variables to Set Data Range

**Exporting Data of Barrier Trials into Excel Files**

Choose [Tools]-[Export Barrier Trial Data] to export barrier trial data to an Excel file.

**New Starter Views**

New starter views added:

1. All Distance WQP Stat (please see All Distances WQP Stats View)
2. Jockey Change Stat (please see Jockey Change Statistics)

© Welkin Systems Limited

3. Common Statistical Figures

**New History/Statistics Views**

1. * Jockey Change Performance (View opened by double-clicking on statistical columns of the starter view "Jockey Change Stat". See All Distances WQP Stats View for details)
2. * Distance WQP Stats (View opened by double-clicking on statistical columns of the starter view "All Distance WQP Stat". See All Distances WQP Stats View for details)
3. * Stable Quinella Place
4. * 1st Section Leading

**New MCL Functions**

- HrStartIDOn(), JrStartIDOn(), SrStartIDOn()
- HrPastIncident( ), JrPastIncident( ), HrPastIncident( )
- DateOfRace( )
- ComingDate( )

## 2.2 What is New in RaceMate 10.5

RaceMate has been updated to version 10.5 in September 2010. Here are the changes since version 10.0.

**Boxes for Storing Calculated Values**

Seven storage boxes are added. You can store any value into one of the boxes. All views share the same storage boxes. That means once you store a value into a storage box, the value in it can be accessed in any views. Values storage in a starter view can be accessed in a history/stat view as well. Storage box values will be kept until you quit RaceMate. Open the help page for Box( ) to learn more.

**Standard Deviation and Variance in History/Stat Views**

Two new statistical items "Standard Deviation" and "Variance" are added into the History/Stat Views.

**Data on the Speed Power page of the Hong Kong Jockey Club website**

Two sets of functions are added to return data on the Speed Power page of the Hong Kong Jockey Club website. See HrPowerFigure( ) and HrPowerComments( ) on the "New or Changed MCL Functions" section below.

**Penetrometer and Clegg Hammer Readings**

A [Penetrometer..] button is added onto the top right the "Race Data" form. Click on it to show the Penetrometer and Clegg Hammer Readings of the course. You may also use the new MCL functions HrPenetrometer( ), JrPenetrometer( ) and SrPenetrometer ( ) to retrieve the values into a column of your views.

© Welkin Systems Limited

### The Jockey Silks Form

The Jockey Silks form was added into the Hong Kong edition in January 2010. Now Jockey Silks form is also functional in Macau edition. You may open the jockey silks form from the race data form and barrier trial form in both the Hong Kong and Macau edition. Open the Jockey Silks help page for detailed operations of the form.

### New or Changed MCL Functions

Box( )

Function to store values into and retrieve values from storage boxes.

HrBTrial( ), JrBTrial( ), SrBTrail( )

Besides returning date and group of trials, now the function is able to return other data of a barrier trial, e.g. draw and finishing position.

HrPowerFigure( ), JrPowerFigure( ), SrPowerFigure( )

The functions return the speed figures on the Speed Power page of the Hong Kong Jockey Club website. In Macau edition, these functions return zeros.

HrPowerComments( ), JrPowerComments( ), SrPowerComments( )

The functions return the comments on horses on the Speed Power page of the Hong Kong Jockey Club website. In Macau edition, these functions return blank values.

HrPenetrometer( ), JrPenetrometer( ), SrPenetrometer( )

The functions return the Penetrometer and Clegg Hammer Readings of the race course.

## 2.3 What is New in RaceMate 10.0

RaceMate is upgraded to version 10 in Sep 2009.

### Unicode Handling Enhancement

Unicode support has been added into RaceMate since version 9.5. You have been able to copy Chinese characters from RaceMate to other Unicode applications like Excel and Word. However once your Chinese characters are captured as Unicode in RaceMate, you can only paste them into other Unicode applications. If you paste them into RaceMate itself, they will become garbage. In this new version, we have enhanced the copy and paste feature. Now Chinese text copied in RaceMate can be pasted into any applications including RaceMate itself without any problem.

### New Debugging and Testing Form for MCL Functions and Expressions

Similar to an Excel spreadsheet, a RaceMate "view" can do column calculations, statistics, and lookup. However even though you can do almost everything with MCL functions, once there is a bug in an expression, you will see a .Null. value and will not get much information about the problem. We have added an expression debugging tool into version 10. To learn more about this tool, please see the page MCL Debugging.

© Welkin Systems Limited

### Importing Data from Excel Spreadsheets into Views

In prior versions, you can only input your comments and ratings directly into RaceMate. However you may not do it in other computers without RaceMate installed. In version 10 or later, user ratings and comments can be input into Excel sheets in a computer without RaceMate. You can then import the data of these files into your RaceMate.

If the file to be imported contains a primary key (in the required format), you can do better do data import. However primary key is not necessary for importing. Please see Importing View Data From Excel Files to learn more about the new feature.

### Exporting View Data to Excel Sheets

In prior versions, we can only export view data race by race via the Exporting or Copying Grids Cells technique. Now, in version 10 or later, you can export view data of multiple races to Excel sheets in one time. Data will be calculated whenever necessary before exporting. See page Exporting View Data to learn more about the view data export feature.

Now you can:

- Export a continuous number of races in one time
- Add a special primary key to the exported files. It makes future data import in RaceMate more accurate.

See the page Format of Exported Excel Files to learn more about the format of the exported file.

### Transferring View Data Between Two RaceMates

In prior version, generally speaking, the only way to transfer data from one RaceMate to another is using the backup and restore functions of RaceMate. However this is an all-or-none approach. You will overwrite any changes (including view structure and data) in the target RaceMate. The new export and import features of RaceMate enable data merging. You can now transfer only the parts that have been changed to another RaceMate.

### Copy All Grid Cells into Windows Clipboard

Now you can copy all grid cells into the Windows clipboard. See Exporting Contents of Grids in Forms for more information.

### New Input Columns for Overseas Races

New input columns are added into the overseas record page of the Horse form. Now you can input your comments into these column for future reference.

© Welkin Systems Limited

## 2.4 What is New in RaceMate 9.8

RaceMate is upgraded to version 9.8 on 9 Mar 2009. Most of the new features are for the Hong Kong edition. If you are running the Macau edition at the moment, you may not be able to open some links at the moment because all links provided are linked to the Hong Kong Jockey club materials. Some of the features will be added into the Macau edition later.

**Multimedia MCL Functions**

A series of multimedia features have been added into RaceMate many years ago. However, such features can't be accessed by MCL and can't be added into a view.

The following list shows a handful of new Multimedia functions. Normally these new functions will only display a date (some with group or section numbers) in character format showing the time the event happens. Because this is a date in character format, they can't be used for date calculations. When you double-click on the date characters, the Jockey Club video will be played. If you want to display a string other than the date, you can pass a value to corresponding argument. Please click on the links below to see a detailed description about the syntax of the functions.

- HrRaceReplay( ), JrRaceReplay( ), SrRaceReplay( )
- HrBTVideo( ), JrBTVideo( ), SrBTVideo( )
- HrPreRace( ), JrPreRace( ), SrPreRace( )
- HrPostRace( ), JrPostRace( ), SrPostRace( )
- HrRacingFocus( ), JrRacingFocus( ), SrRacingFocus( )

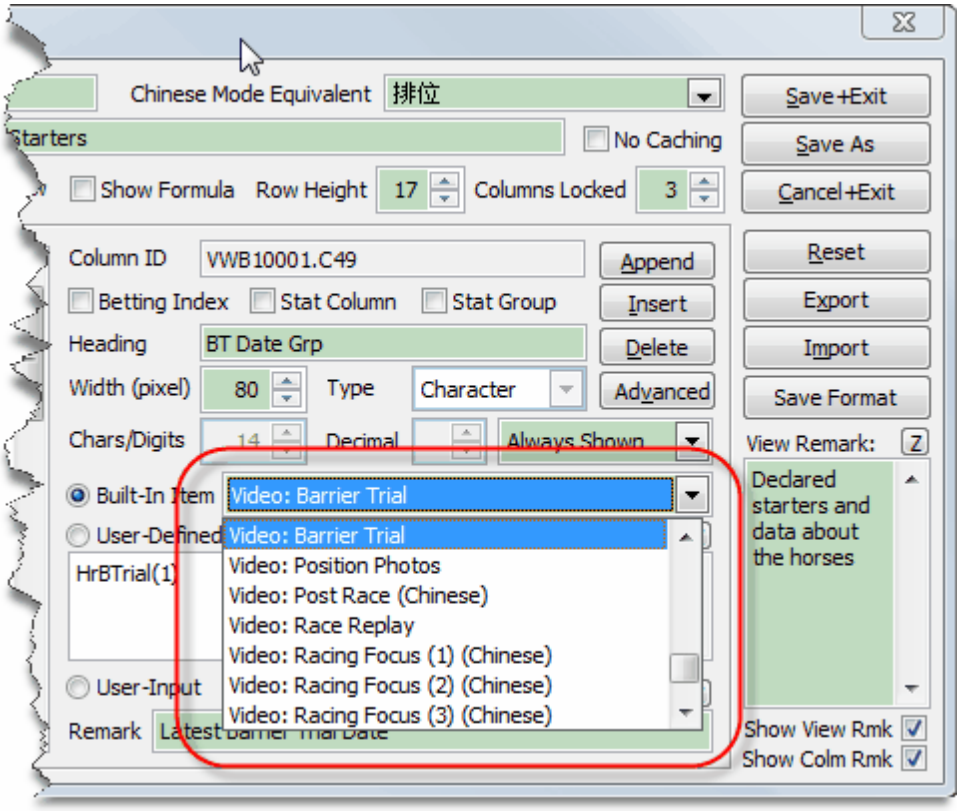
The following functions will be able to open the Race Running Positions Pictures.

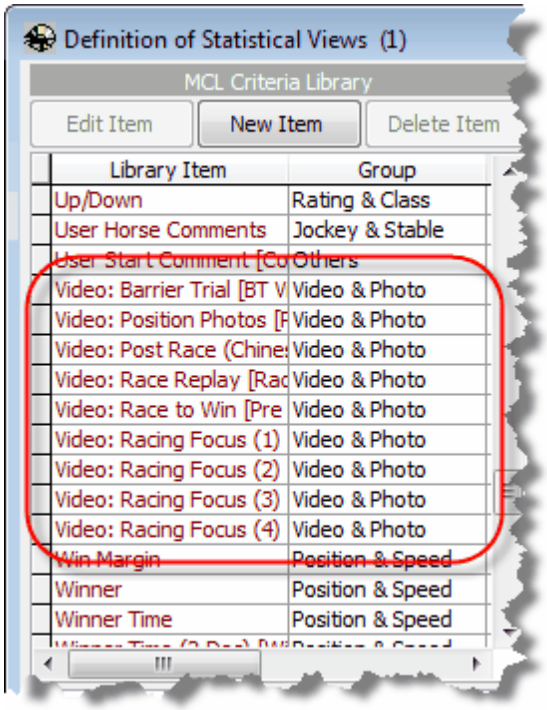
- HrPosPhoto( ), JrPosPhoto( ), SrPosPhoto( )

**New Built-In Columns**

In the new version, all the new multimedia functions are added into the built-in columns list. If you want to add any of them into your own view, you may simply choose the one appropriate.

© Welkin Systems Limited




**Barrier Trial Functions**

In the latest version of RaceMate we have added the following new functions to our MCL. All the following functions will return a string show the date and group of barrier trials. If you don't pass any argument, the function will return the latest barrier trial before the race. You may pass a number into the function to get earlier trials. Please

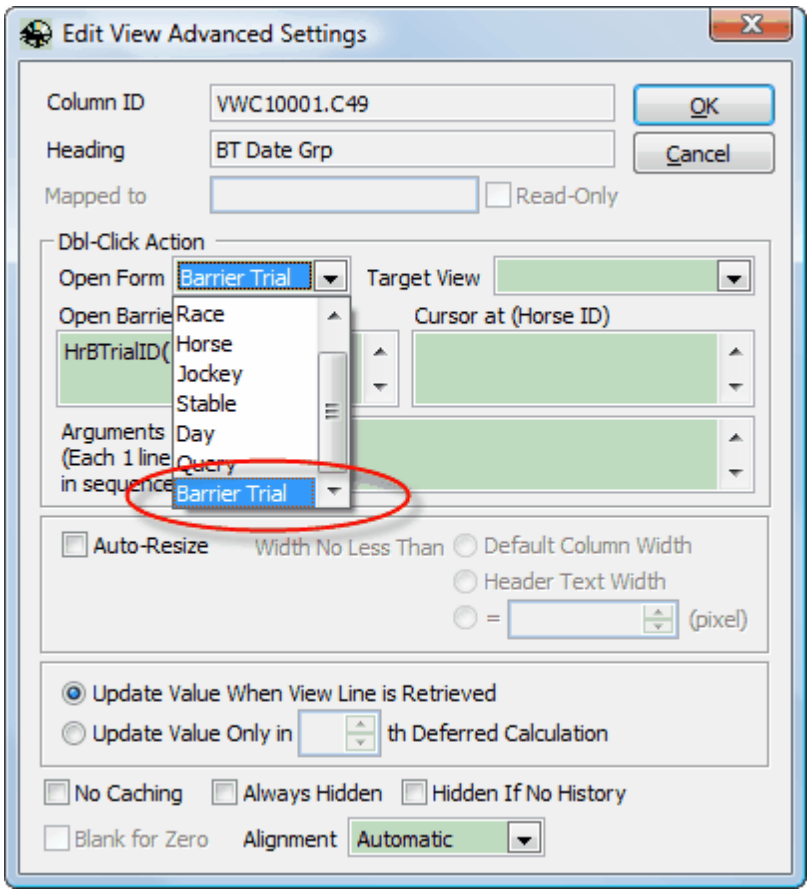
© Welkin Systems Limited

note that both HrBTrial() and HrBTrial(1) returns the same value. Double-clicking on the date and group number of the column will open the corresponding barrier trial form.

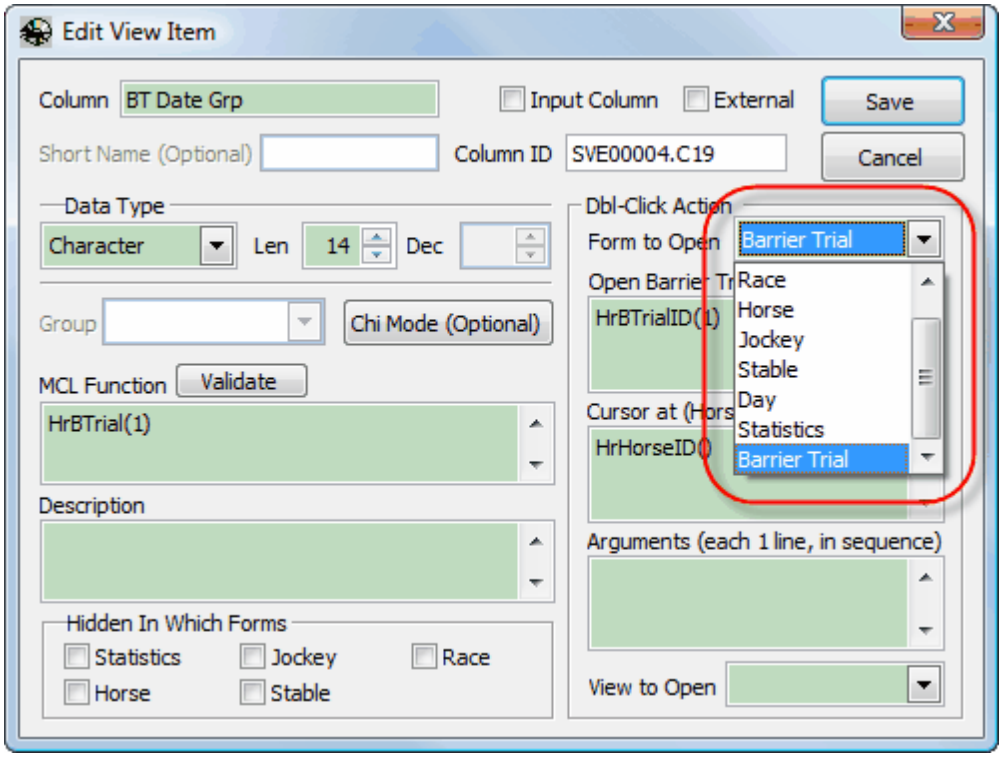
- HrBTrial( ), JrBTrial( ), SrBTrial( )

### Chaining to Barrier Trial Form

In prior versions of RaceMate, we can only chain to forms for Race, Horse, Jockey, Stable, Day and Query. Now you will have one more choice: Barrier Trial form.




© Welkin Systems Limited




### Use [Alt] to Open Related Forms

When you double-click on the multimedia function columns in a view, the corresponding video will be played. If you want to open the related forms side by side, you may hold down the [Alt] key and double-click on the column once again.

| Function | Video/Photo Double-Click to Open | Form Opened by [Alt] Double-Click |
|---|---|---|
| HrRaceReplay( ) | Race Replay Video | Race Data Form |
| HrBTVideo( ) | Barrier Trial Video | Barrier Trial Form |
| HrPreRace( ) | Race to Win Video | Race Data Form |
| HrPostRace( ) | Post Race Video (Chinese) | Race Data Form |
| HrRacingFocus( ) | Racing Strategy Video (Chinese) | Race Calendar Form |
| HrPosPhoto( ) | Race Running Photos | Race Data Form |

## 2.5 What is New in RaceMate 9.5

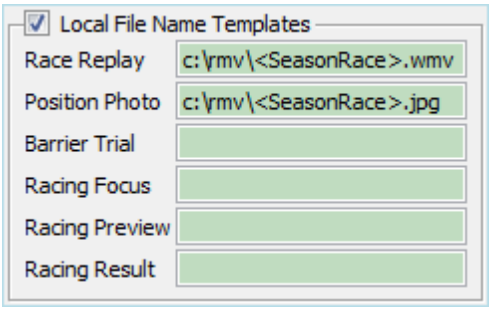
### Transferring Multimedia Files from Websites into Local Drive

Many users like downloading video files and position pictures from websites into local hard disk to get fast playing and viewing. You can then keep your files even if they are removed from the web. If you are doing similar things, you no longer need to manually create links to the files that you saved.

You can open the "Options and System Information" form from the [Tools] menu. At the right top of the form you will see an area for entering a template for each file type as follows. Once you have followed the rules on the page Saving Web Video/Photo into Disk and correctly enter corresponding templates into the boxes in the figure below

© Welkin Systems Limited

(you can leave some of them blank), RaceMate's default links or buttons will point to these local files rather than the one on websites.




Examples

| Template | Click On | File Opened |
|---|---|---|
| c:\Replay\<SeasonRace->.ivr | Race Replay link or button of race 7 on 29/4/2007 (season 06-07) | c:\Replay\060559.ivr |
| c:\File\HK<Year-><Month><Day>_<RaceNo>.ivr | | c:\File\HK070429_07.ivr |
| c:\Video\HK-<DateStr->-<RaceNo>.ivr | | c:\Video\HK-070429-07.ivr |

See the page Saving Web Video/Photo into Disk to learn how to create a template.

How can I save a web video as a local file?

There are many tools which help you do the job. For example the latest version of RealPlayer and FlashGet. Both of them are free of charge. Please read their manuals for detailed operations.

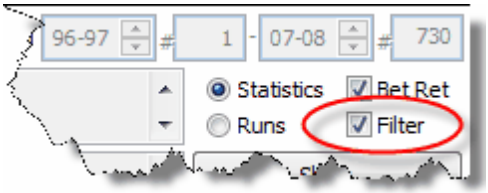
## Changes in User Interface

Stakes of a Horse

On the Horse form, you will find a new text box displaying the stake money of the horse.

Filter On/Off on History/Stat Page

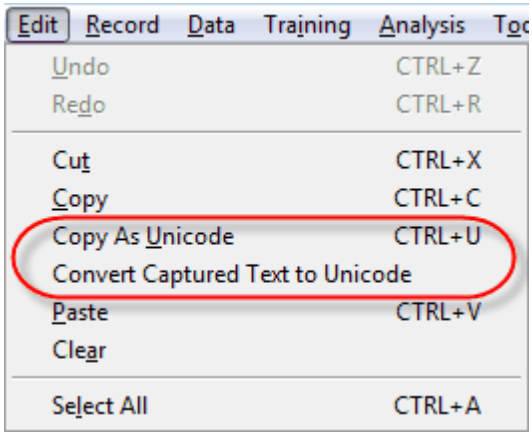
On the Comprehensive Statistics and Record Query form and the History/Stat page of the Horse, Jockey, and Stable forms, a new [Filter] check box has been added. You may use it to turn on or off filters.




© Welkin Systems Limited

### Copy and Paste Chinese From RaceMate into Unicode Software (e.g., Word & Excel)

If you have ever copied and paste Chinese characters from RaceMate into the latest versions of Word or Excel, you would have found that the Chinese characters pasted became interesting symbols. It is because RaceMate stores and displays Chinese as double-byte characters (DBCS) while many other applications store Chinese as Unicode. Since RaceMate 9.5, you can use the following new menu options to copy and paste Chinese.




| Option | Description |
|---|---|
| Copy As Unicode | Copy selected characters into the clipboard as Unicode, so that you can paste the text into other Unicode applications. |
| Convert Captured Text to Unicode | If you have copied text as DBCS (rather than Unicode) characters via the Ctrl+C keys, you may change the DBCS text in clipboard into Unicode via this menu option, so that the text can then be pasted into other Unicode applications. |

### New MCL Functions

The following new MCL functions were added. Please open their help pages for details.

- HrSectMargins( ), JrSectMargins( ), SrSectMargins( )
- HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( )
- HrSectMargLensX( ), JrSectMargLensX( ), SrSectMargLensX( )
- HrSectPositions( ), JrSectPositions( ), SrSectPositions( )
- HrSectTime( ), JrSectTime( ), SrSectTime( )
- HrCurRating( ), JrCurRating( ), SrCurRating( )
- HrJockeyEngAbbr( ), JrJockeyEngAbbr( ), SrJockeyEngAbbr( )
- HrNetLoadDiff( ), JrNetLoadDiff( ), SrNetLoadDiff( )

### MCL Functions Updated

1) HrTwCount( ), JrTwCount( ), and SrTwCount( )

New symbols for the cTemplate argument of HrTwCount( ), JrTwCount( ) And SrTwCount( ):

| Symbol | Return |
|---|---|

© Welkin Systems Limited

| | |
|---|---|
| P | Spelling |
| N | No activity (including spelling) |
| I | No activity (excluding spelling) |

2) 'DayX-DayY' (Syntax 2) of HrTwCount( ), JrTwCount( ) And SrTwCount( )

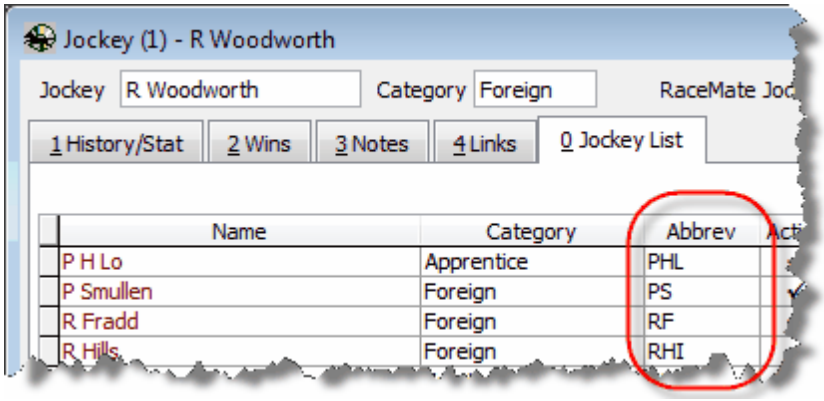
In old versions, the numbers are from large to small (e.g., 40-10, where 40 means 40 days ago). RaceMate 9.5 changes the sequence to small to large (e.g., 10-40) making it consistent with that in the functions HrActivity( ), JrActivity( ) and SrActivity( ).

3) HrActivity( ), JrActivity( ) and SrActivity( )

In old versions and Chinese mode, in the results of HrActivity( ), JrActivity( ) and SrActivity( ), jockey names were shown as the first Chinese characters of jockeys, resulted in many duplications. In order to solve the problem, in version 9.5, it will be shown in Jockey Club's jockey ID (returned by the new function HrJockeyEngAbbr( ))

### Jockey Abbreviations in Jockey List of Jockey Form

An Abbreviation (Abbrev) column has been added into the jockey list of the jockey form. In the Hong Kong edition, the abbreviations are basically the Jockey IDs in various documents of the Hong Kong Jockey Club.




### New Built-In Columns

In Race, Horse, Jockey, Stable and Comprehensive Statistics forms, the following new items were added for selecting.

- Horse Section Time
- Horse Section Margins
- Horse Section Margin Lengths
- Positions (with FP)
- Loading Different (After Adj)
- International Rating
- Latest Horse Rating
- Prior FPs of Top 4 Horses (Only in Starter Views)

### New Structures of Views

The structures of the "Declaration" and "Position" views of the Race form and the

© Welkin Systems Limited

"Basic Columns" view of the "History/Stat" page of the Horse, Jockey, Stable and Comprehensive Statistics forms have been revised. New columns for the new sectional data from the Jockey Club were added. (In order to see the change, existing users must reset their corresponding views. However, resetting a view will remove all your changes to the view. If you want to keep your change, please don't do it.)

## 2.6 What is New in RaceMate 9.0

Welkin has done significant improvements in RaceMate's new version 9.0. In order to fit in with the latest operating system Windows Vista, we have redesigned part of RaceMate's structure. There are also many exciting feature enhancements in version 9.0. The main objective of this version upgrade is to make RaceMate a unique professional tool for organizing race information and getting useful betting hints. Below is a highlight of part of the new features.

### Changes Related to Windows Vista

Microsoft has implemented radical changes in the security design of Windows Vista. Part of the program code of RaceMate was rewritten in order to make the software fit in with the new operating system.

New Default Installation Folder

- RaceMate is a dynamically updating software. Users can download new data everyday from RaceMate's data server. During data updates, program files may also be renewed in order to enhance the software's features and to deal with occasional changes implemented by the Jockey Club. Therefore RaceMate must be installed into a writable folder. One major change of Windows Vista is the new read-only default of the "C:\Program Files" folder. Because of this change, RaceMate can no longer reside in "C:\Program Files". The default installation folder of RaceMate has been changed to "C:\RaceMate". You may change C: to other drive letters if you want RaceMate resides in another drive.
- This default installation folder change will not affect existing RaceMate users running RaceMate under Windows XP or earlier.

In Case You Upgrade Your Windows

In case you are going to upgrade your Window to Vista from an earlier version, no matter you are doing an in-place upgrade or moving RaceMate into a new Windows Vista computer, please follow the steps in the pages About UpgradingWindows and Moving or Uninstalling RaceMate.

### New Features about Statistics and Record Filtering

Unless you regard horse racing as pure gambling, you should know that statistics and record filtering are the most important actions for winning good betting returns. The new statistics and record filtering power make RaceMate a professional unique tool of data analysis for getting best betting results.

Unlimited Number and Complexity of Criteria

- The "Race Query and Comprehensive Statistics" form has been completely rewritten. In old versions, only one level of "AND" criteria can be formed. In version 9.0, you can create any number of AND or OR brackets in your criteria in unlimited levels (a criteria tree with any number of AND and OR nodes).

© Welkin Systems Limited

- In old versions, there is a set of predefined data columns for showing the results of all queries and statistics. You can neither get rid of any of them nor add a new one. In version 9.0, you can customize your queries and statistics results. You can create different views for different queries.
- While keeping the user interface simple, a number of new criteria items have been added into the criteria list of the Race Query and Comprehensive Statistics form. You can now do almost any statistics under any grouping by simple mouse clicks.

Statistical Views and Criteria Library

- In prior versions, only a predefined set of criteria are available for choosing. In version 9.0 you can define new criteria by inputting MCL functions. You can even add your criteria into the new MCL Criteria Library for future use.
- In old versions, the system will not keep any record of criteria selections that you make on the query form. That means you have to reselect all the criteria next time you want to carry out the same statistics. In version 9.0, you can save what you selected as views. The views will be available in the view list of the query page for future access.

Views in Horse, Jockey, and Stable Forms

- Views are now available in the horse, jockey, and stable forms. You will see a couple of built-in views in these forms. You may also design your own views to retrieve the data that you need and calculate the statistics that you want. A new Stat Views page will replace the History page.
- You can now create user input columns in the new Stat Views for entering comments and ratings.
- You can edit input columns of other views. In other words, an input column can be shared by more than one view.

Sharing and Hiding Views

- As described above, a view is now shared by different forms. Therefore you will have a centralized location for editing a view and any new changes to the view will take place in all forms.
- The new view sharing design brings statistics and filter features (which are only available in the Comprehensive Statistics form in order versions) to the Horse, Jockey and Stable data forms. This will provide you with new angles for studying statistical results.
- In case a view is not suitable for a particular form, you may hide the view from that form. Whenever needed, you can show the view again on the form. In case you save a built-in view as a new view for editing, you may then hide the original built-in view from the view list of the Stat View page.

**Differences between New and Old Views**

In RaceMate 8.0 or earlier, views (starter views) can only be found in the Race Data form. In the new version, views (statistical views) are available in Comprehensive Statistics, Horse, Jockey and Stable forms. Please note that the views in the Race Data form of RaceMate 9.0 are still old type views. Actually there are many important differences between these two types of views. The major differences are listed as follows.

Views in Different Languages Mode

Each starter view in the Race Data form is only created for a particular language

© Welkin Systems Limited

mode. You will not see a starter view created in English when you are in Chinese language mode, and vice versa. On the contrary, the new statistical views can be for both language modes. There is a checkbox on the view property editing form which enables bilingual support of the view.

Column Names and Column ID Numbers

- Duplicated column names are not allowed in starter views while they are legal in new statistical views. This design is to facilitate the new tree structure of criteria, particularly "OR criteria groups". That means a criteria title can appear multiple times in the view structure table.
- Each column in a statistical view is assigned a column ID number. In the new version, the Column( ) function will accept either column ID numbers (e.g., Column (15)) or column names (e.g., Column('FP')) as argument. Since now column names are not necessarily unique, we recommend users to use column ID numbers in the Column( ) function in new statistical views. Column ID numbers are guaranteed to be unique and will not be changed when you switch language modes while names of columns will generally be different in different language modes.

Differences about Built-in Views

- There are two types of built-in starter Views: modifiable and read-only, while there is only one type of built-in views in the new statistical views of version 9.0: modifiable with auto-reset. The changes that you make to an old modifiable built-in starter view will be permanent while changes made to a new built-in statistical view will be lost (auto-reset) when you restart RaceMate. If you want to keep your change to a built-in statistical view, you can save the changed view structure to a new view name. You can then make further changes to the view without worrying about losing your work.
- The reason we don't want to keep the changes to a built-in view made by a user is that we will enhance and update built-in views on a regular basis and we don't want to overwrite users' change during the updates. The reason we allow temporary changes is that user can change statistics grouping of a built-in view to compare the results of grouping changes (drill-down and roll-up). We will prepare video demos about this technique later.

Pause before Data Retrieval

On the data forms with a new view selection list, changing views by single-clicking on a view name will not immediately retrieve data. You will only see the data grid is shaded. You must then click on the [Load Data] button to start data retrieval. The reason for this design is that in a starter view of a Race Data form, there are limited number of records because there are only less than 14 horses in a race, while in other data forms number of records can be over thousands. Loading thousands of records with statistics can be time consuming. Therefore we place a pause before data retrieval so that you can adjust view criteria and data range before data loading. If you really want immediate data retrieval without pause, please double-click on a view name rather than single-click on it.

Relocating Columns

In new statistical views, you will not be able to relocate a column by directly dragging it into a new location. In order to move it, you need to open the structure of the view in the "Definition of Statistical Views" form and use the arrow buttons at the right bottom of the form to relocate a column line.

© Welkin Systems Limited

## 2.7 What is New in RaceMate 8.0

### Database Engine Upgrade

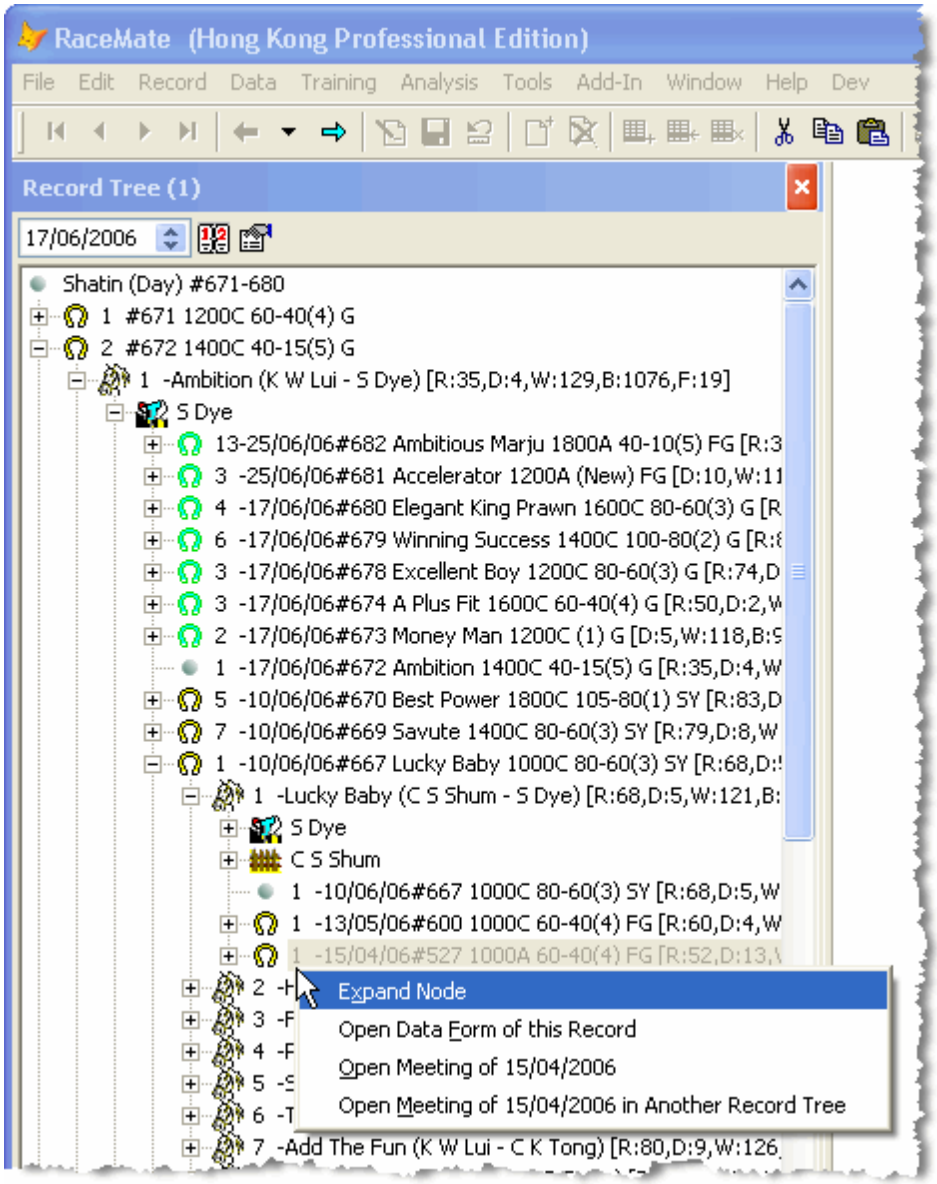
The database engine of RaceMate 8.0 has been upgraded to Microsoft Visual FoxPro 9.0 Service Pack 1. Microsoft has fixed many significant bugs in this Service Pack 1 making this release of Visual FoxPro the most stable version.

### Record Tree

The Record Tree is an innovative design which combines features of the Calendar, Race, Horse, Jockey and Stable forms into an expandable and collapsible data tree. Both backward (past) and forward (future) records can be shown in the tree. You can follow detailed history of each horse, jockey and stable by expanding the nodes (unlimited levels) of the tree. Opening two Record Trees side by side is an excellent way to compare two different records scenarios. You can even configure the numbers of past and future records shown in the tree to meet you own requirements.

The Record Tree is actually a data navigator for opening different data forms. For example, clicking on a race record will open the Race Data form, clicking on a horse record will open the Horse Data form, and so on.

© Welkin Systems Limited


RaceMate (Hong Kong Professional Edition)
File Edit Record Data Training Analysis Tools Add-In Window Help Dev
Record Tree (1)
17/06/2006
Shatin (Day) #671-680
1 #671 1200C 60-40(4) G
2 #672 1400C 40-15(5) G
1 -Ambition (K W Lui - S Dye) [R:35,D:4,W:129,B:1076,F:19]
S Dye
13-25/06/06#682 Ambitious Marju 1800A 40-10(5) FG [R:3
3 -25/06/06#681 Accelerator 1200A (New) FG [D:10,W:11
4 -17/06/06#680 Elegant King Prawn 1600C 80-60(3) G [R
6 -17/06/06#679 Winning Success 1400C 100-80(2) G [R:
3 -17/06/06#678 Excellent Boy 1200C 80-60(3) G [R:74,D
3 -17/06/06#674 A Plus Fit 1600C 60-40(4) G [R:50,D:2,W
2 -17/06/06#673 Money Man 1200C (1) G [D:5,W:118,B:9
1 -17/06/06#672 Ambition 1400C 40-15(5) G [R:35,D:4,W
5 -10/06/06#670 Best Power 1800C 105-80(1) SY [R:83,D
7 -10/06/06#669 Savute 1400C 80-60(3) SY [R:79,D:8,W
1 -10/06/06#667 Lucky Baby 1000C 80-60(3) SY [R:68,D:
1 -Lucky Baby (C S Shum - S Dye) [R:68,D:5,W:121,B:
S Dye
C S Shum
1 -10/06/06#667 1000C 80-60(3) SY [R:68,D:5,W
1 -13/05/06#600 1000C 60-40(4) FG [R:60,D:4,W
1 -15/04/06#527 1000A 60-40(4) FG [R:52,D:13,
Expand Node
Open Data Form of this Record
Open Meeting of 15/04/2006
Open Meeting of 15/04/2006 in Another Record Tree
7 -Add The Fun (K W Lui - C K Tong) [R:80,D:9,W:126

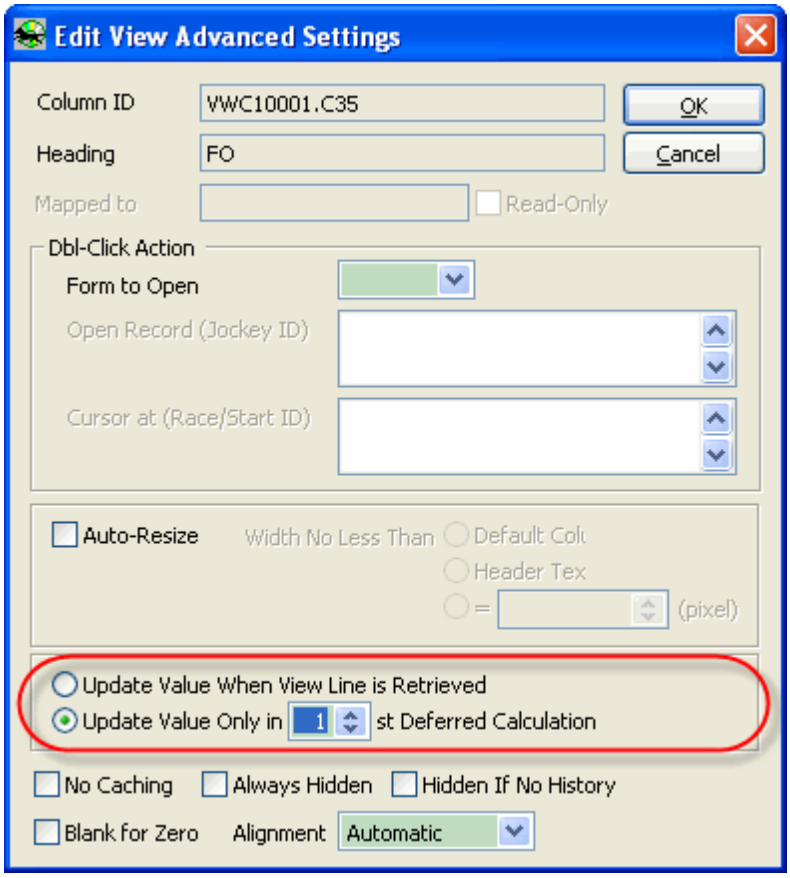
**Deferred-Calculation of View Columns**

RaceMate carries out view calculations row by row. The calculation result of a cell may be wrong if it is a function referencing other rows which are yet to be updated. In order to get correct results, you need to recalculate the view by clicking on the [Recalc] button. For some complicated views, you may need to recalculate even more than once. In RaceMate 8.0, you can now selectively set a number of columns carry out calculations only after the base calculation is finished. We call such a new feature Deferred-Calculations. Deferred-Calculations is much more efficient than doing simple recalculations in the following ways:

1. Each column of the view will carry out calculation only once. Deferred-calculating columns will not be updated in the initial base calculation.
2. If there is more than one deferred-calculating column, you may define the calculation order of each deferred-calculating column

With this sophisticated new feature, you will no longer bother about recalculations and will always get the results that you want.

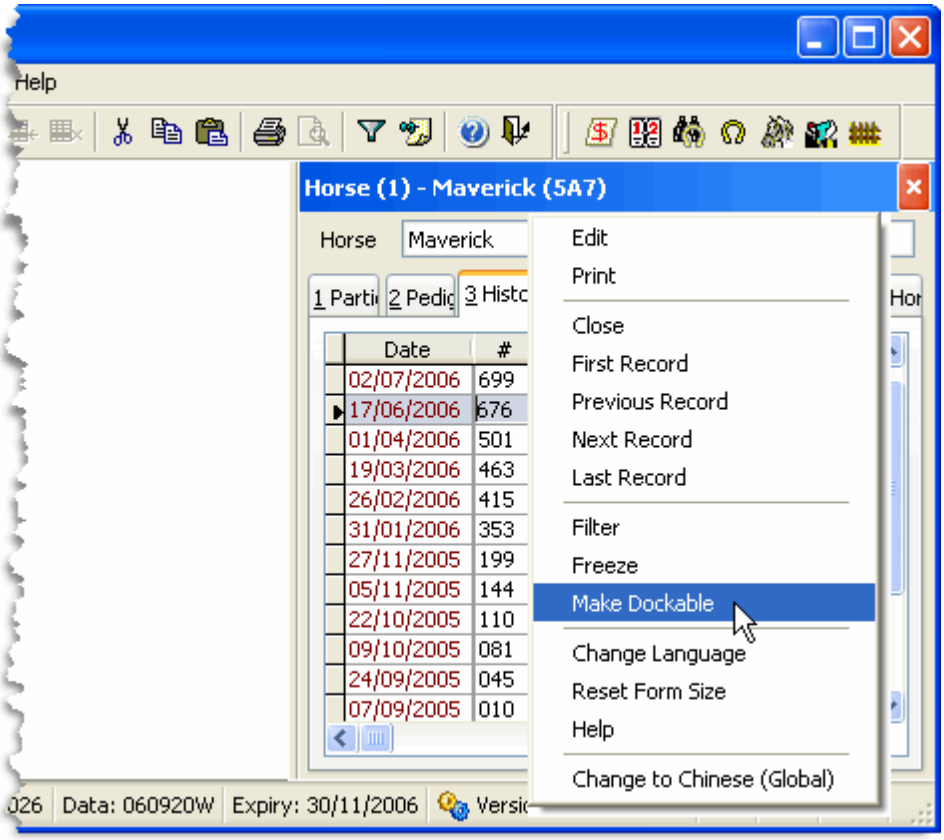
© Welkin Systems Limited




### Form Docking

In RaceMate 8.0, main data forms can be made dockable. When a form is made dockable, You may stick (dock) the form to a border of the RaceMate application window. A docked form will won't be blocked by another maximized form.

© Welkin Systems Limited




### Moving a Form Out of RaceMate's Application Window

When a form is made dockable, you can not only dock the form at a RaceMate border, but also move it out our RaceMate's application window. Nowadays many computer users are equipped with dual (or even triple) monitors. In RaceMate 8.0 you can place data forms in different monitors. The work space for RaceMate will dramatically be enlarged.

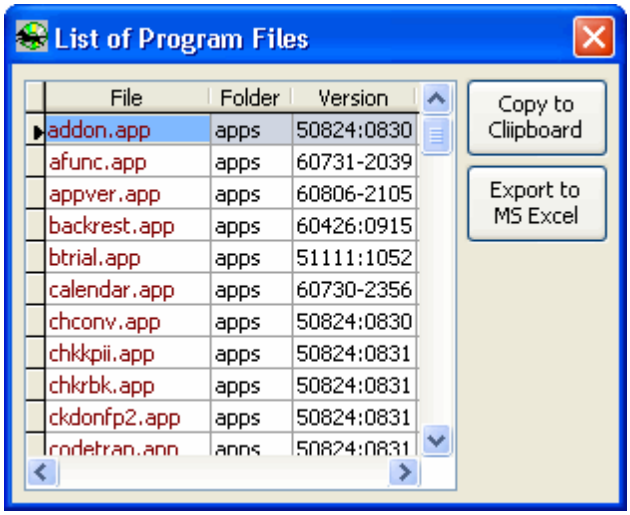
### Export Data of All Grids to Microsoft Excel

Microsoft Excel is renowned for its powerful data analysis power and is the favourite tool for many computer users. In previous versions of RaceMate, only a couple of forms are equipped with the "export to Microsoft Excel" feature. In RaceMate 8.0 all data grids are exportable to Excel. If you have never tried to create charts and pivot tables in Excel for data analysis, you are seriously recommended to investing your time to learn these wonderful data tools.

### Version Details of Program Files

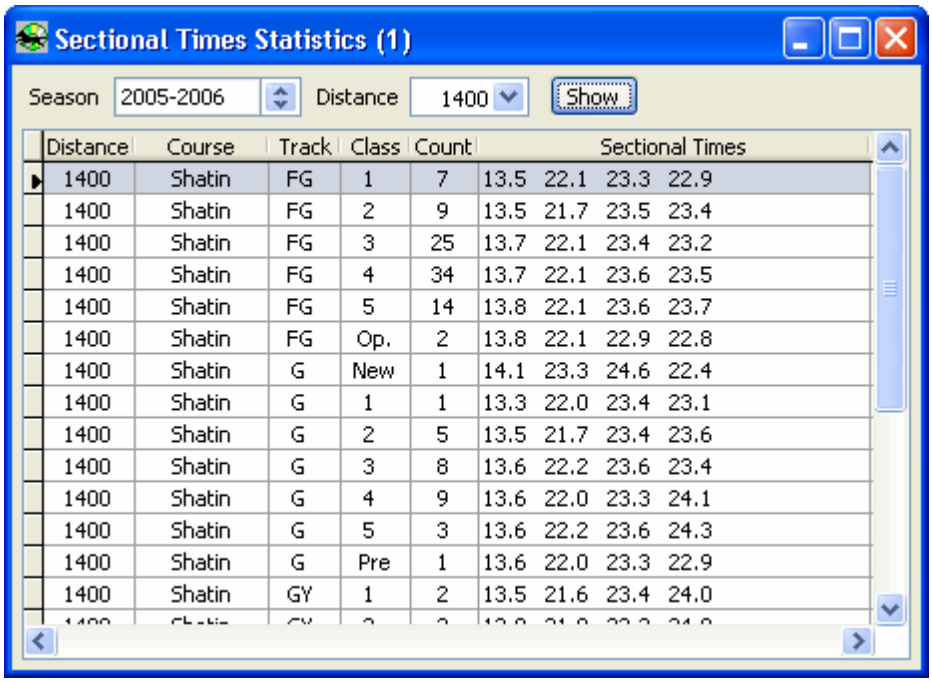
Double-clicking on the version label of the status bar will show a List of Program Files with detailed version information. By sorting on the Version column, you can know which program files are recently get updated. The [Export to MS Excel] button lets you export this information into an Excel spreadsheet. In case your RaceMate show irregular behaviours, this file will be helpful to RaceMate developers in Welkin to locate the problems in your RaceMate installation.

© Welkin Systems Limited




### Sectional Times Statistics Form

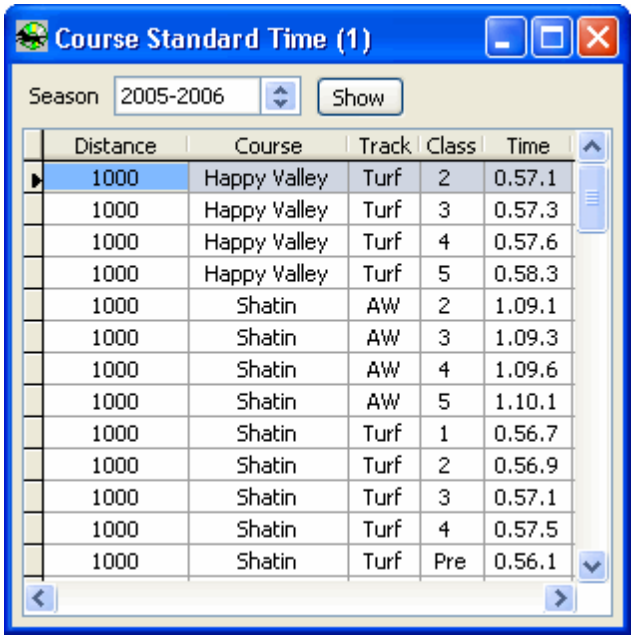
A new Sectional Times Statistics form is added into RaceMate 8.0. The form shows average sectional times of all races in all combinations of distance, course, track, and class. You may carry out multiple columns sorts or filter a particular distance in the form.




### Course Standard Time Form

A new Course Standard Times form is added into RaceMate 8.0. The form shows the average standard running time of all races in all combinations of distance, course, track, and class. You may carry out multiple columns sorts.

© Welkin Systems Limited




| Distance | Course | Track | Class | Time |
|---|---|---|---|---|
| 1000 | Happy Valley | Turf | 2 | 0.57.1 |
| 1000 | Happy Valley | Turf | 3 | 0.57.3 |
| 1000 | Happy Valley | Turf | 4 | 0.57.6 |
| 1000 | Happy Valley | Turf | 5 | 0.58.3 |
| 1000 | Shatin | AW | 2 | 1.09.1 |
| 1000 | Shatin | AW | 3 | 1.09.3 |
| 1000 | Shatin | AW | 4 | 1.09.6 |
| 1000 | Shatin | AW | 5 | 1.10.1 |
| 1000 | Shatin | Turf | 1 | 0.56.7 |
| 1000 | Shatin | Turf | 2 | 0.56.9 |
| 1000 | Shatin | Turf | 3 | 0.57.1 |
| 1000 | Shatin | Turf | 4 | 0.57.5 |
| 1000 | Shatin | Turf | Pre | 0.56.1 |

**New MCL Functions**

HrCountBy( ), JrCountBy( ), SrCountBy
HrSumBy( ), JrSumBy( ), SrSumBy( )
HrAverageBy( ), JrAverageBy( ), SrAverageBy( )

This series of new functions magically empowers the already strong MCL statistical functions. The functions do statistical calculations and break down calculation results in a single function call. Results are composed of indefinite number of statistical items and returned as character strings. The following screen capture shows a view with 3 statistical columns which break down the number of runs and wins of each horse. Please see corresponding help topics for the detailed usages of these new functions.

| Horse | Runs: HrCountBy(,'HrDistance()') | Wins: HrCountBy('HrFp()=1','HrDistance()') | Wins: HrCountBy(,'HrDistance()',,'For HrFP()=1') |
|---|---|---|---|
| Wise Lover | 1000(1) 1200(3) 1400(6) 1600(1) | 1000(0) 1200(1) 1400(1) 1600(0) | 1200(1) 1400(1) |
| Whiplash | 1000(2) 1200(2) 1400(4) | 1000(0) 1200(0) 1400(2) | 1400(2) |
| Lifeline Rapi | 1000(3) 1200(3) 1400(17) 1600(2) | 1000(0) 1200(0) 1400(2) 1600(0) | 1400(2) |
| Progressing | 1000(7) 1200(8) 1400(6) 1650(12) | 1000(0) 1200(0) 1400(0) 1650(0) | |
| Master Majo | 1400(1) | 1400(0) | |
| Lucky Dansk | 1000(2) 1200(6) 1400(10) 1650(1) | 1000(0) 1200(1) 1400(2) 1650(0) | 1200(1) 1400(2) |
| Siameric Te | 1200(6) 1400(1) 1650(2) | 1200(3) 1400(0) 1650(0) | 1200(3) |
| Lucky Drago | 1000(1) 1200(1) 1400(16) 1600(1) 1650(2) | 1000(0) 1200(0) 1400(2) 1600(0) 1650(0) | 1400(2) |
| Honour Supr | 1000(4) 1200(24) 1400(6) 1600(1) 1650(3) | 1000(0) 1200(5) 1400(0) 1600(0) 1650(0) | 1200(5) |
| Super Snipp | 1400(1) | 1400(0) | |
| Spokesman | 1000(1) 1200(12) 1400(11) 1600(2) 1650(6) | 1000(0) 1200(0) 1400(1) 1600(0) 1650(2) | 1400(1) 1650(2) |
| Classabove | 1400(3) | 1400(0) | |
| Multi-win Spi | 1200(3) | 1200(0) | |
| Global Way | 1400(2) | 1400(0) | |

StdTime( )

The new StdTime( ) function returns the standard running time of races of a particular combinations of distance, course, track, and class.

Argument for Future Records

The nBackRace argument of most MCL functions now supports negative numbers for retrieving data of future races. For example, HrFp(-1) will return the finishing position of next race (relative to the underlying race) while HrFp(1) returns the finishing

© Welkin Systems Limited

position of last race.

Miscellaneous

There are many improvements, in terms of both functionality and efficiency, and bug fixing in the MCL language in the RaceMate 8.0.

**2007-02-01**

1. A new built-in column "W,Q,P Numbers - Same Jockey" is added. You may insert this new column into any views.
2. The following columns are added into the "WQP Statistics" view:
   - o Jockey
   - o WQPU(Tot)-Jockey

   In order to see this new change, you must reset the view by clicking on [Modify]-[Reset]. Please note that resetting a view will wipe out all your prior change made to the view.
3. A new filter "Same jockey" is added into the filter list of the Starter Views page.

**2006-12-13**

New MCL function:

HrRaceTime( ), JrRaceTime( ), SrRaceTime( )

**2006-12-06**

Three new buttons added into the Race Calendar form: [Prev Month], [Current Mtg], and [Next Month]

**2006-11-20**

New button [Go To Current Meeting] added on the Record Tree form

**2006-11-20**

New auto-hide feature for the Record Tree

**2006-11-13**

1. New columns in the History page of Horse, Jockey and Stable forms
2. Use the new HrDistTrack( ) function (for better sorting operation) for the distance column in the History page of Horse, Jockey and Stable forms

**2006-11-09**

1. New user interface in the Starter Views page of the Race Data form:

   Dropdown list of filters and filter on/off checkbox

   If you want to see only records which satisfies a criterion, you may set a filter on a starter view. In the past, filter is set by clicking on the "Filter" button on the RaceMate toolbar. Since filter setting is one of the most important steps of studying a race, we have put a filter dropdown list at the top of the Starter Views

© Welkin Systems Limited

page together with a checkbox for turning on and off a filter. Setting a filter is now simply one or tow mouse clicks.

Relocation of controls for past records

The dropdown list for changing the track of history and the spinner for setting the number of history rows are now moved to the page top at the left of the filtering controls.

Balloon tips for view remarks

Now when you move your mouse over a view name in the view list, you will see a balloon tip showing the remark for the view. You may turn this tip on or off in the Edit View form.

Balloon tips for column remarks

Now when you move your mouse over the header of a column, you will see a balloon tip showing the remark for the column. You may turn this tip on or off in the Edit View form.

Newly added built-in view shown in red

In order to draw users' attention to newly added built-in views, whenever a new built-in view is added into RaceMate, its name will be shown in red in the view list. Once you have opened the view, its color will return to black.

2. Saving an existing view as a new view:

   Now you can save an existing view as a new view. This is the fastest way of creating a new view from an existing view. You may then edit the newly created view without changing the structure of an existing view.

3. Read-only built-in views

   In the past, all built-in views are editable. You can add or remove columns from a built-in view without any restriction. However, this was a problem for developers to update built-in views even if they found mistakes or improvement needs in the views. It is because any update to a built-in view may overwrite changes made by users or break the formulae added by users. A new type of built-in view, namely "ready-only built-in view", is therefore introduced. A read-only built-in view is not editable. However, users may save the view as a new editable view for their own customization needs.

4. New "Best Distance" view

   The newly added "Best Distance" view is the first read-only view of RaceMate. It shows the winning percentage of horses, jockeys, and stables in different distances together with the date of their latest wins. Users can get an idea of which distance is best for a contestant. This view utilizes of the powerful new v8.0 functions HrCountBy( ), JrCountBy( ) and SrCountBy( ). Please see the help pages for these functions for details.

5. New built-in columns

   * Horse Winning% with Dates by Distance
   * Jokcey Winning% with Dates by Distance
   * Stable Winning% with Dates by Distance
   * Distance+Track

**2006-11-06**

The following functions now support "Enhanced Syntax":

© Welkin Systems Limited

- HrCountBy( ),JrCountBy( ), SrCountBy( )
- HrSumBy( ),JrSumBy( ), SrSumBy( )
- HrAverageBy( ),JrAverageBy( ), SrAverageBy( )

The Enhanced Syntax addition makes the above functions the most powerful MCL functions of RaceMate. Now you can:

1. carry out multiple statistics in a single pass of past record scanning
2. customize the format of return strings
3. change the order of breakdown groups
4. define a summary session in the return string

Please read the help pages for the above functions for details.

**2006-10-31**

New MCL functions added: HrDistTrack( ), JrDistTrack( ), SrDistTrack( )

**2006-10-16**

New MCL functions added:

HrClassGroup( ),JrClassGroup( ), SrClassGroup( )

**2006-10-13**

Application status bar is updated back to version 8.0 style.

**2006-10-03**

- In the Results page of the "Race Query and Comprehensive Statistics" form, a new column displaying time of the last section is added.
- New function ListStdDev( ) for calculating Standard Deviation value is added.

**2006-09-23**

In the Allocation page of the Race Data form, horses not confirmed to start will be in brackets.

**2006-09-20**

New MCL functions HrDvd( ),JrDvd( ), SrDvd( ) added for retrieving all dividends data in RaceMate database.

**2006-09-14**

New MCL functions added:

- Join( )
- ListSort( )
- ListSortVal( )
- ListSortDesc( )
- ListsortDescVal( )

New built-in columns added:

- 3 Fastest Times (Dist, Course, Going)
- 3 Fastest Times Average

© Welkin Systems Limited

- Horse Wins by Draw
- Horse Wins by Distance
- Horse Wins by Jockey
- Jockey Wins by Draw (12months)
- Jockey Wins by Distance (12months)
- Stable Wins by Draw (12months)
- Stable Wins by Distance (12months)
- Stable Wins by Jockey (12months)

**2006-09-13**

Application status bar is reverted to version 7.0 style.

**2006-09-05**

- New columns added into the Race Search form
- New columns added into the Index page of the Race Data form

**2006-08-24**

New built-in columns added:

- Time Taken before Final Section

New comprehensive statistics item added:

- Time Taken before Final Section

## 2.8 What is New in RaceMate 7.0

**Why 7.0 Instead of 6.0?**

RaceMate for DOS 6.5 (released in about 1999) was the last DOS version RaceMate. A number of RaceMate users are still keeping this memorable DOS RaceMate in their hard drive. In order to avoid confusion, the Windows version of RaceMate will be upgraded from version 5.0 to 7.0, skipping the version number 6.0.

**Database Engine Upgrade**

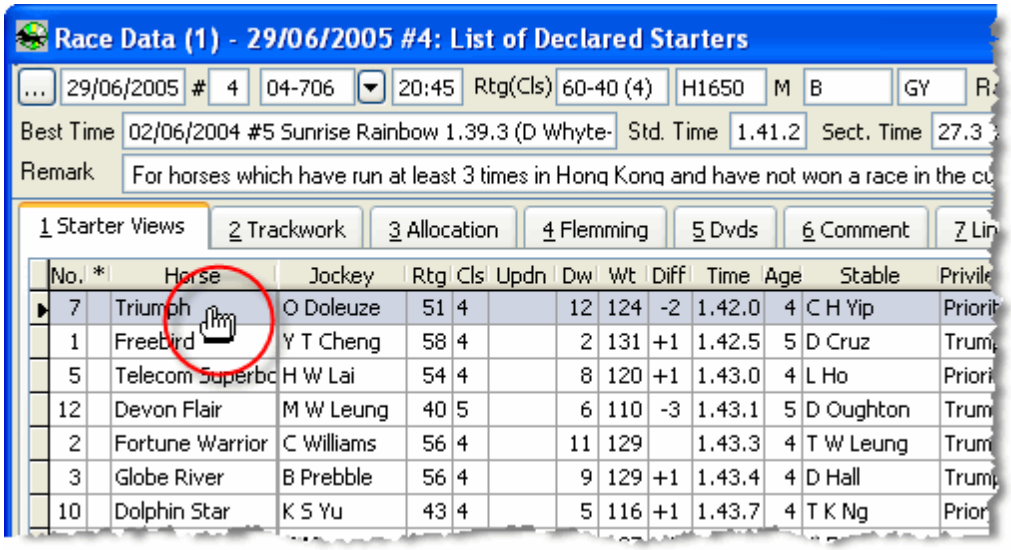
The database engine of RaceMate 7.0 has been upgraded to Microsoft Visual FoxPro 9.0, which is the latest and most stable version of Microsoft Visual FoxPro since its first release in 1995. Microsoft has fixed quite a number of bugs in this latest release. You will experience far less system abnormalities in RaceMate 7.0. Microsoft Visual FoxPro is the fastest desktop database engine available as at this writing. No other databases engine can match its lightning data query speed.

**User Interface Enhancement**

Visual cue for hyperlinking fields

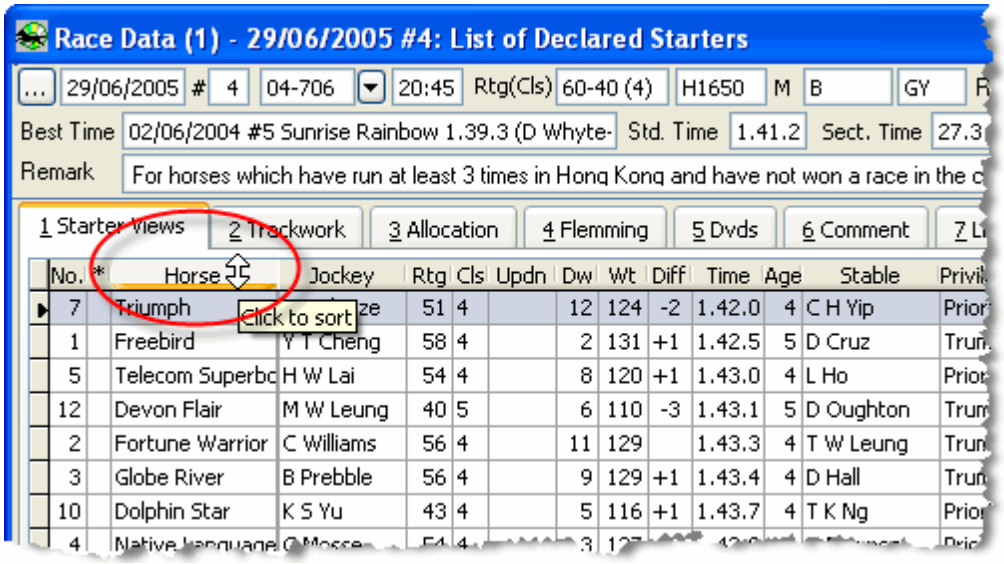
Whenever you double-click on a horse name column in a view, the Horse Data form will be opened. We call the such a column a hyperlinking column. In RaceMate 7.0, when your mouse hangs over a hyperlinking column, the mouse pointer will turn to an overlapping hands icon, which prompts for double-clicking.

© Welkin Systems Limited


Race Data (1) - 29/06/2005 #4: List of Declared Starters
29/06/2005 # 4 04-706 20:45 Rtg(Cls) 60-40 (4) H1650 M B GY
Best Time 02/06/2004 #5 Sunrise Rainbow 1.39.3 (D Whyte- Std. Time 1.41.2 Sect. Time 27.3
Remark For horses which have run at least 3 times in Hong Kong and have not won a race in the cu
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Comment 7 Lin
No. * Horse Jockey Rtg Cls Updn Dw Wt Diff Time Age Stable Privile
7 Triumph O Doleuze 51 4 12 124 -2 1.42.0 4 C H Yip Priorit
1 Freebird Y T Cheng 58 4 2 131 +1 1.42.5 5 D Cruz Trum
5 Telecom Superbo H W Lai 54 4 8 120 +1 1.43.0 4 L Ho Priori
12 Devon Flair M W Leung 40 5 6 110 -3 1.43.1 5 D Oughton Trum
2 Fortune Warrior C Williams 56 4 11 129 1.43.3 4 T W Leung Trum
3 Globe River B Prebble 56 4 9 129 +1 1.43.4 4 D Hall Trum
10 Dolphin Star K S Yu 43 4 5 116 +1 1.43.7 4 T K Ng Prior

Visual cue for sortable columns

Many table columns of RaceMate can be sorted by clicking on their column headers. In RaceMate 7.0, when you put your mouse over the heading of a sortable column, the mouse pointer will change to a double arrow icon, which cues you that the column can be sorted in either ascending or descending orders.


Race Data (1) - 29/06/2005 #4: List of Declared Starters
29/06/2005 # 4 04-706 20:45 Rtg(Cls) 60-40 (4) H1650 M B GY
Best Time 02/06/2004 #5 Sunrise Rainbow 1.39.3 (D Whyte- Std. Time 1.41.2 Sect. Time 27.3
Remark For horses which have run at least 3 times in Hong Kong and have not won a race in the c
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Comment 7 Li
No. * Horse Jockey Rtg Cls Updn Dw Wt Diff Time Age Stable Privil
Click to sort
7 Triumph 51 4 12 124 -2 1.42.0 4 C H Yip Prior
1 Freebird Y T Cheng 58 4 2 131 +1 1.42.5 5 D Cruz Trum
5 Telecom Superbo H W Lai 54 4 8 120 +1 1.43.0 4 L Ho Prior
12 Devon Flair M W Leung 40 5 6 110 -3 1.43.1 5 D Oughton Trum
2 Fortune Warrior C Williams 56 4 11 129 1.43.3 4 T W Leung Trum
3 Globe River B Prebble 56 4 9 129 +1 1.43.4 4 D Hall Trum
10 Dolphin Star K S Yu 43 4 5 116 +1 1.43.7 4 T K Ng Prior

**View Definition Enhancement**

In the new version, you can assign a form to be hyperlinked when a user-defined column is double-clicked. The hyperlinking property of the user-defined column is defined in the Advanced Settings form, which will be shown when you click on the [Advanced] setting. You may define the following items on the form:

1. Form to Open: Choose the form that your want to open here.
2. Open Race (RaceID): Input an MCL function returning the ID of the target race, horse, jockey, or stable which you want to show on the hyperlinked form. RaceMate will evaluate the function and use the returned ID for opening the corresponding record in the target form.
3. Cursor at (Horse ID): Input an MCL function returning the ID of a horse. RaceMate will try to locate the horse in past record table in the target form.

© Welkin Systems Limited

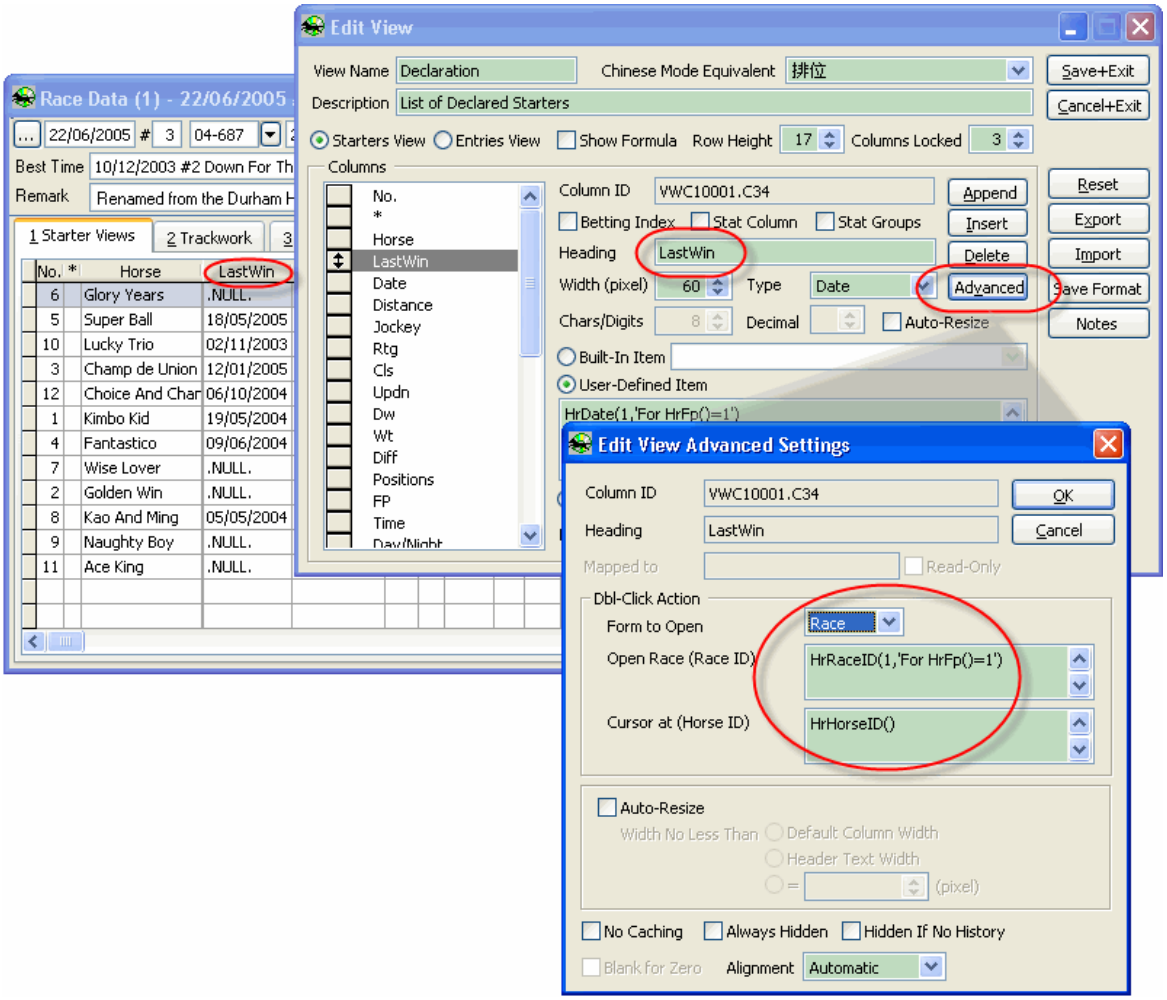
In the example in the following figure, a user-defined column LastWin is defined in the Declaraction view. The column returns the last winning date of each horse using the formula:

```
HrDate(1,'For HrFp( )=1')
```

When you double-click on the column, RaceMate will open a new Race Data form and open the race record corresponding to the race ID returned by the following function:

```
HrRaceID(1,'For HrFp( )=1')
```

After the race record is open, RaceMate will then highlight the horse that you double-click on (returned by the function HrHorseID( )) in the past record list.

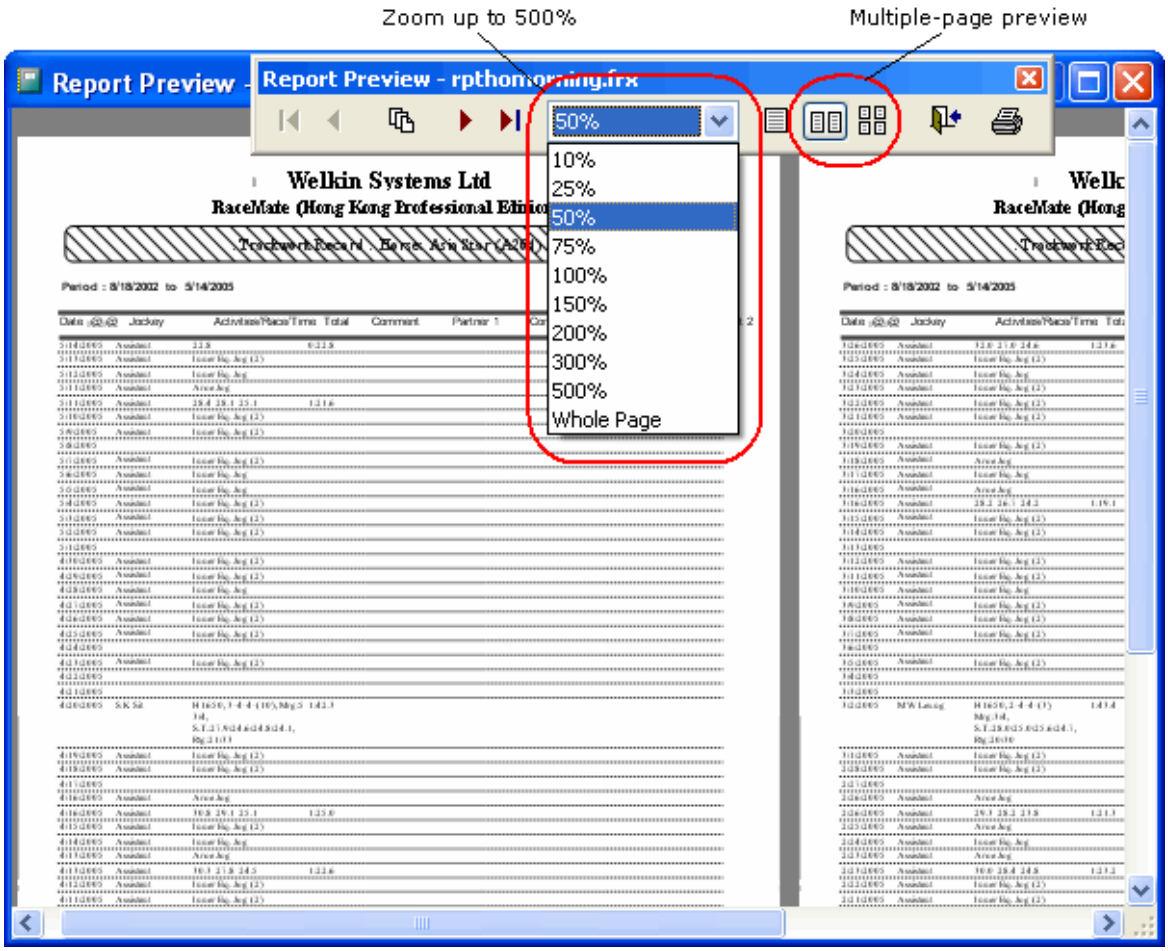



## New Report Engine

Zoom percentage

New zoom percentage options are available in the dropdown list of the Report Preview toolbar. You may now zoom a report up to 500%.

© Welkin Systems Limited

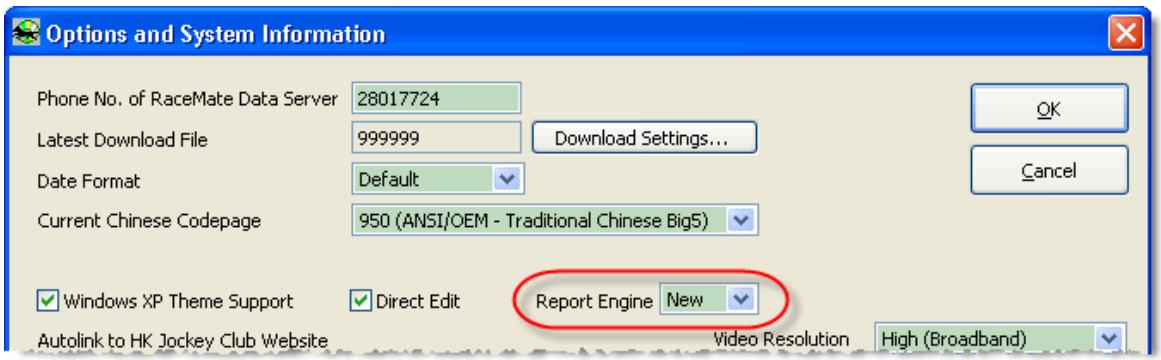



Multiple-page preview

You may now display more than one page in the new report preview form by clicking on the report preview toolbar. Viewing a considerably track work record of a considerably long period may be helpful in finding special training pattern horses. If you have a large monitor, you will find this new feature valuable.

Reverting to old report engine

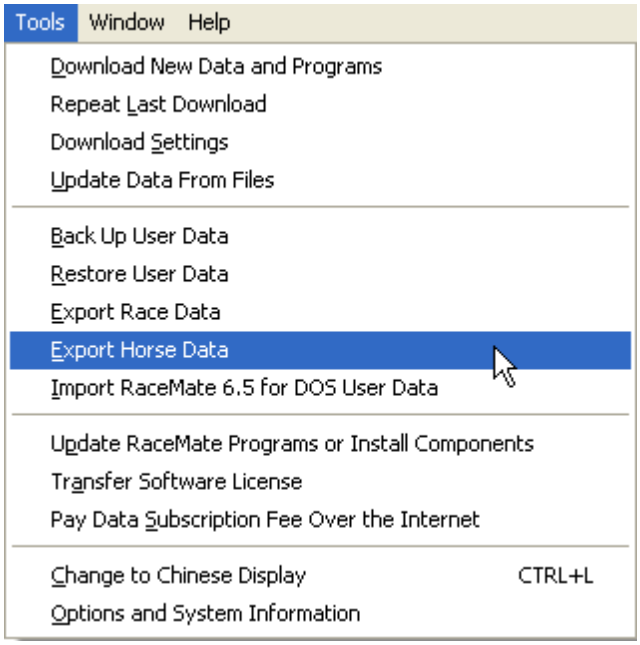
The new report engine is powerful but runs in a much lower speed. If you feel comfortable with the old report engine and do not need the new functionalities which the new report engine provides, you may revert the report engine to its previous version in the RaceMate Options page as show in the following figure.




© Welkin Systems Limited

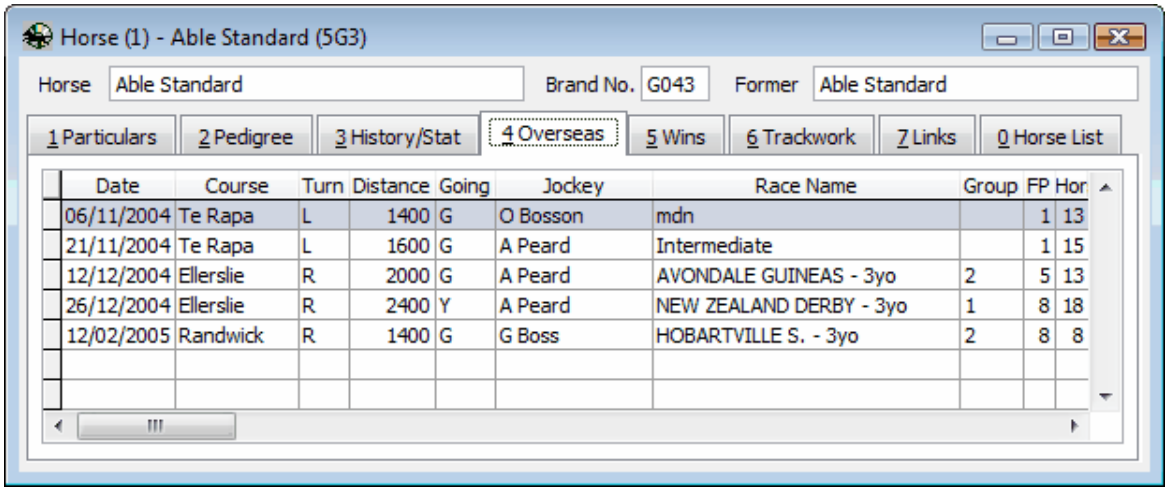
### Exporting Horse Information

You may now export data of horses into Microsoft Excel. Choose [Export Horse Data] from the [Tools] menu to get access to the new function.




### Overseas Records

A new "Overseas" page is added into the Horse data form. The page shows the overseas racing record of the horse before it comes to Hong Kong.




| Date | Course | Turn | Distance | Going | Jockey | Race Name | Group | FP | Hor |
|---|---|---|---|---|---|---|---|---|---|
| 06/11/2004 | Te Rapa | L | 1400 | G | O Bosson | mdn | | 1 | 13 |
| 21/11/2004 | Te Rapa | L | 1600 | G | A Peard | Intermediate | | 1 | 15 |
| 12/12/2004 | Ellerslie | R | 2000 | G | A Peard | AVONDALE GUINEAS - 3yo | 2 | 5 | 13 |
| 26/12/2004 | Ellerslie | R | 2400 | Y | A Peard | NEW ZEALAND DERBY - 3yo | 1 | 8 | 18 |
| 12/02/2005 | Randwick | R | 1400 | G | G Boss | HOBARTVILLE S. - 3yo | 2 | 8 | 8 |

### Multiline to Character Column Type Conversion Support in Views

In previous versions of RaceMate, when you modify a view and change the field type of a column from "Multiline" text type to "Character" type, the content of the column will be lost. In the new version, change the column type from Multiline to Character or vise versa will keep all the content.

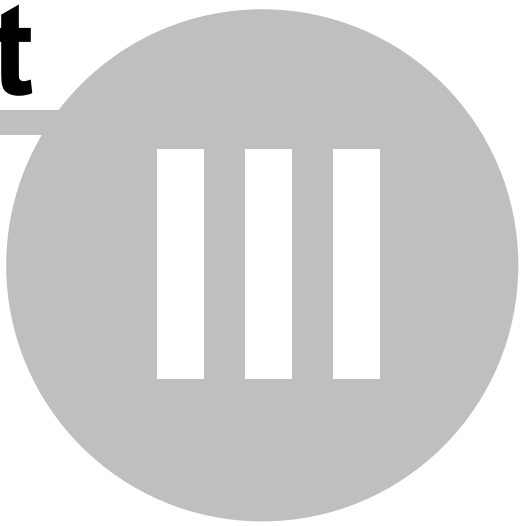
### New Setup

© Welkin Systems Limited

The setup program for RaceMate 7.0 has completely rewritten. A setup program for RaceMate 5.0 Hong Kong Edition is about 170M in size while that for RaceMate 7.0 is only about 35M. Whenever a use buys a new computer or finds it RaceMate data corrupted, he may download a new setup program form RaceMate's website. The effective compression of the new version dramatically reduces the download time.

© Welkin Systems Limited

# Part



# 3 Data Update

You can download race and track work data from RaceMate's Data Server through the Internet or dialing up to our dataserver. That means you can download data from anywhere in the world.

We suggest you to download via the Internet rather than dialing up to our data server. It is because:

1. Downloading through the Internet is much more faster
2. Dial-up connection to our data server is subject to our telephone lines availability. If there are a number of people already connecting to the server at the same time, you may not be able to get through.

**>>> Subtopics of this section <<<**

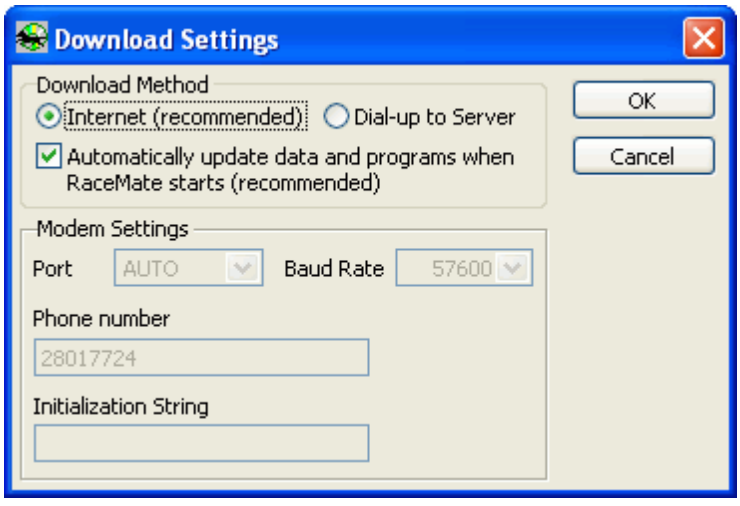
- Connecting to the Internet
- Dial-Up Connection
- Downloading Data
- Repeat Last Download
- Firewall Blocking

## 3.1 Connecting to the Internet

**Download Settings**

Carry out the following download settings to enable downloading data through the Internet:

1. Choose [Tools] – [Download Settings] from the main menu.




2. Select [Internet] on the form.
3. Click on OK.

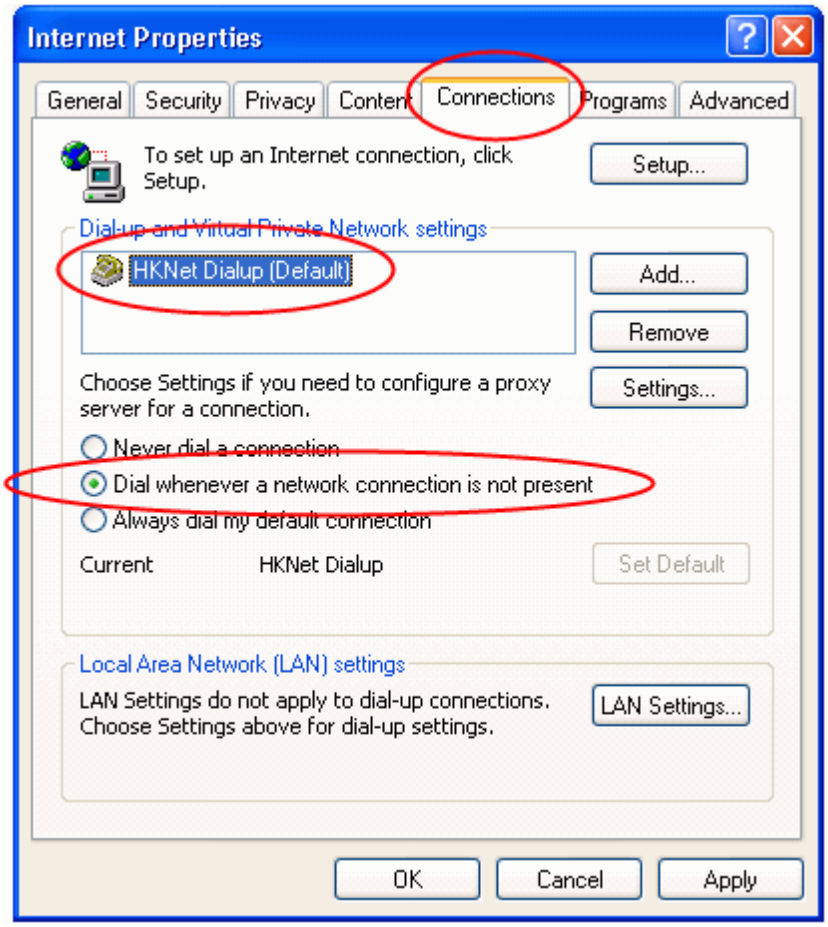
   When you have changed the download method to "Internet", RaceMate will download data through Internet until you change the download method back to "Dial-up to Server". If you are using a broadband connection, the setting is now complete. Otherwise, please read the following section.

**Non-Broadband Users**

© Welkin Systems Limited

If you are using a modem to connect to the Internet, it is better for you to enable the automatic dial-up:

1. Click [Start] – [Settings] – [Control Panel]
2. When [Control Panel] is opened, double-click [Internet Options]
3. Click the [Connections] tab:




4. Select the dial-up network that you would like to use. In this example, "HKNet Dialup" is selected. The name of your connection may be different.
5. Select "Dial whenever a network connection is not present"
6. Click [OK]

**See also**

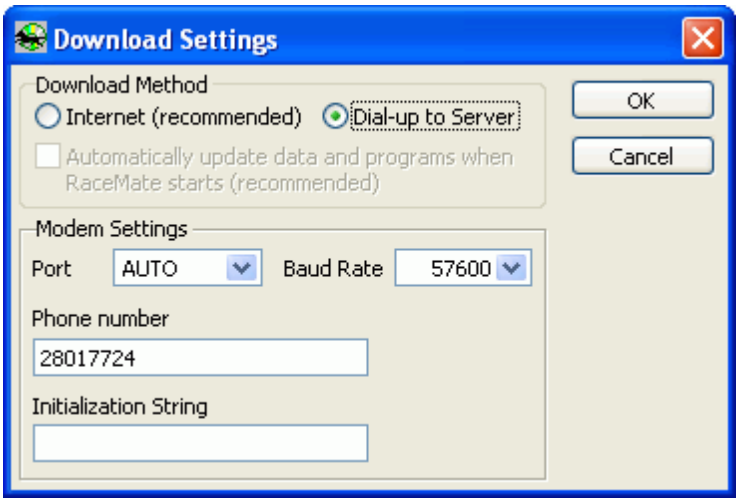
Dial-Up Connection | Downloading Data |

## 3.2 Dial-Up Connection

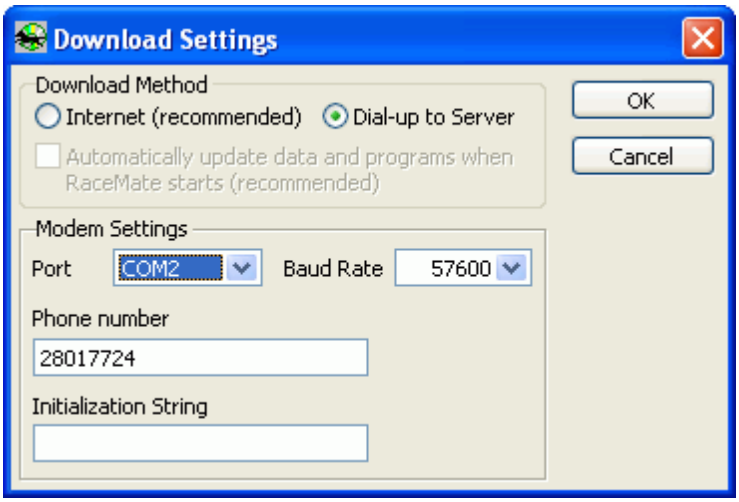
1. Dialing up to RaceMate's data server is an alternative method of data update with lower speed. Since almost all users have an Internet account, this option will only be used as an back-up measure. The telephone lines that we reserve for this option is very limited. We don't guarantee a successful connection from this option.
2. To dial up to the RaceMate data server, your computer must have a modem installed. Follow the steps below to set up the modem:

© Welkin Systems Limited

3. Click [Tools] – [Download Settings] on the menu.
4. When the Download Settings dialog box appears, choose [Dial-up to Server] as the download method.




5. Close the form and try to download data by selecting [Tools] – [Download New Data] from the menu. If there is a connection problem, you may specifically change the Port setting on the Download Settings form to match the COM port number of your modem. In the example in the figure below, COM2 is selected. However, the port of your modem may be different.




You may follow the steps below to check the COM port of your modem:

- Open the Control Panel from the Start menu of your computer. If you are using Windows XP, please choose "Switch to Classic View" at the left top of the Control Panel window.
- Double-click on the Modem icon to open the modem setting form.
- You will see the COM port number of your modem on the "Analysis" page.

6. Click [OK].

**See also**

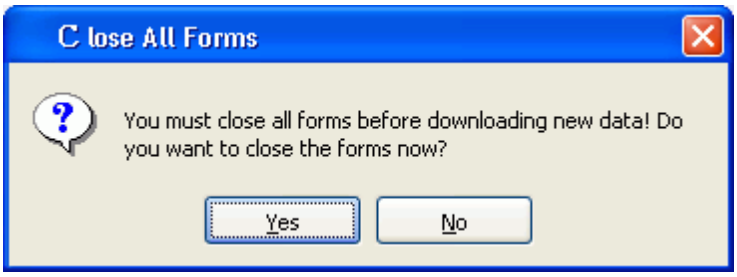
Connecting to the Internet | Downloading Data |

© Welkin Systems Limited

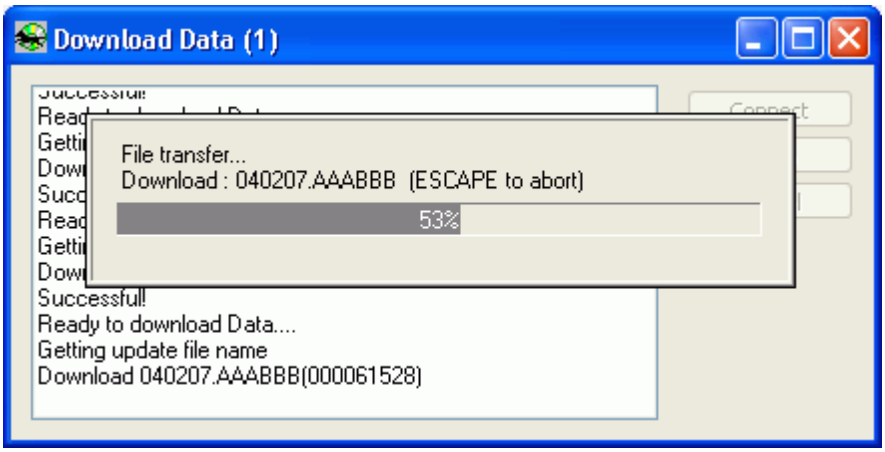
## 3.3 Downloading Data

Before you download data, make sure you have completed the download settings ( Dial-up or Internet). When you want to download new data, please follow the steps below:

1. Click [Tools] – [Download New Data] on the menu.
2. You will see the follow dialog box. Clock on [Yes].




3. The Data Download dialog box appears and RaceMate will automatically connect to the Data Server and retrieve the update files. If the download method is set to Internet, you will see a progress status screen similar to the following diagram:




4. When all the data have been downloaded, the update process will begin. When you see the 'Update finished' message, click OK to complete the whole download process.

**See also**

Connecting to the Internet | Dial-Up Connection |

## 3.4 Repeat Last Download

Once you find a problem in data after you have downloaded and updated your database, you may try selecting [Tools] [Repeat Last Download] to update your data once again.

© Welkin Systems Limited

## 3.5 Firewall Blocking

In order to carry out user registration and data update, RaceMate needs to communicate with Welkin's data servers. Your computer may have a firewall installed. If you see a dialogue box similar to the following, please choose [Unblock] to allow RaceMate communicating with its data server. Please note the following points:

1. The exact message or dialogue you see is dependent on the brand and type of your firewall software. It may be completely different from the following figure. If you don't find the [Unblock] button on the dialogue, please look for other command buttons, option buttons, or checkbox with a caption of similar meaning.
2. Depend on the setting of your firewall, sometimes RaceMate may be block without any message or warning shown on the screen. If you find any problem in user registration or data update, please open the list of blocked software of your firewall and remove RaceMate from the list. Consult your firewall document about how to unblock software.
3. If you don't have a firewall installed or enabled, you will not see similar message.

Windows Security Alert
To help protect your computer, Windows Firewall has blocked some features of this program.
Do you want to keep blocking this program?
Name: RaceMate
Publisher: Welkin Systems Limited
Keep Blocking
Unblock
Ask Me Later
Windows Firewall has blocked this program from accepting connections from the Internet or a network. If you recognize the program or trust the publisher, you can unblock it. When should I unblock a program?

© Welkin Systems Limited

# Part

# 4 Basic UI Techniques

This section describes the basic techniques that you should know when using RaceMate.

**>>> Subtopics of this section <<<**

- Using Menus and Toolbars
- Changing Language of Display
- Hyper-Linking Records
- Incremental Search
- Editing Data
- Handling Forms
- Handling Grids

## 4.1 Using Menus and Toolbars

This section will discuss the functions and operation techniques of RaceMate menus and toolbars.

**>>> Subtopics of this section <<<**

- Menus and Toolbars
- Moving Toolbars

### 4.1.1 Menus and Toolbars

Almost all functions of RaceMate can be get accessed to via menus and toolbars. Some of them are mapped to hot keys. The table below shows some important hot keys:

| Menu Options | Hot Keys |
|---|---|
| Print | [Ctrl + P] |
| Edit | [Ctrl + E] |
| Save | [Ctrl + S] |
| Quit | [Alt + F4] |

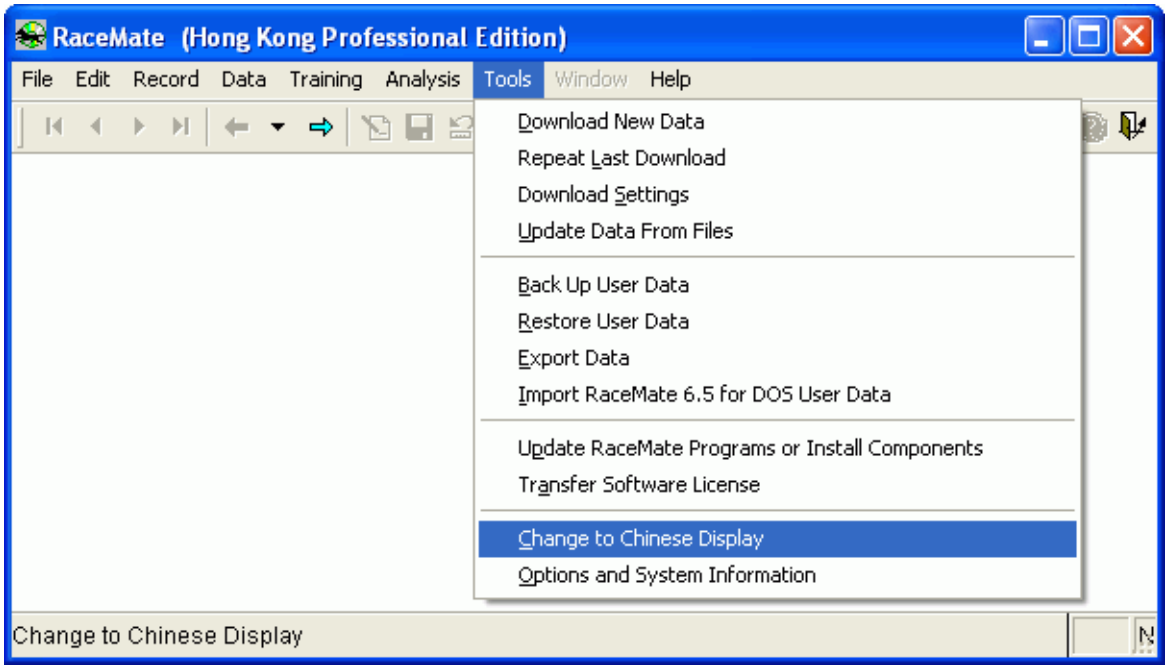
### 4.1.2 Moving Toolbars

RaceMate toolbars can be moved by dragging their handles at the leftmost. You may drag to undock them or dock them to the left, right or bottom of the program windows.

## 4.2 Changing Language of Display

You can change the user interface of RaceMate between English and Chinese. Using Mouse If you are viewing in English and would like to switch to Chinese, click on Tools menu and select Change to Chinese Display. Using Keyboard You can also do it by pressing CTRL+L.

© Welkin Systems Limited


RaceMate (Hong Kong Professional Edition)
File Edit Record Data Training Analysis Tools Window Help
Download New Data
Repeat Last Download
Download Settings
Update Data From Files
Back Up User Data
Restore User Data
Export Data
Import RaceMate 6.5 for DOS User Data
Update RaceMate Programs or Install Components
Transfer Software License
Change to Chinese Display
Options and System Information
Change to Chinese Display

## 4.3 Hyper-Linking Records

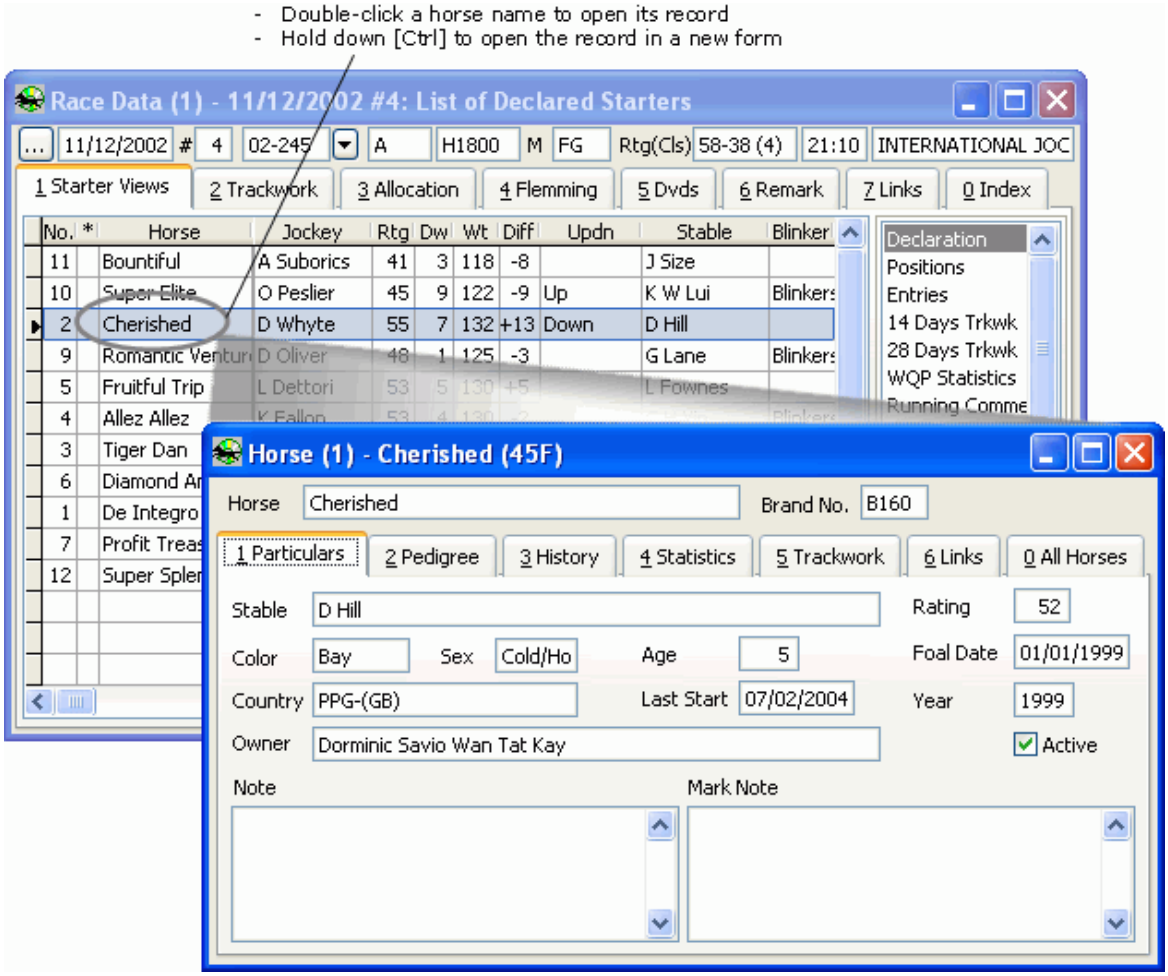
### Hyper-Linking

You may double-click names of Horses, Jockeys, and Stables to open their records or double-click on a race date to open the record of the race.

### Multiple Instances of a Form

In case you double-click on several horses one after one, RaceMate will use a single Horse form the display records of the horses (i.e., the form will always display the horse record that you lastly referenced, and previous record will be replaced). If you want to display the records in separate Horse forms, you may hold down the [Ctrl] key while you double-click.

© Welkin Systems Limited




## 4.4 Incremental Search

### Incremental Search in a Text Column

You may perform an incremental search in almost all columns of the grids of forms. Click into the column that you want to search. Key in the text being searched. Every time you key a stroke, RaceMate will jump to the record line in which the column text best matches (see remark no. 2 below) the keystrokes. The keystrokes that you entered will be highlighted. The following figure shows the process of searching the horse "Blue Stitch".

© Welkin Systems Limited

**Remarks:**

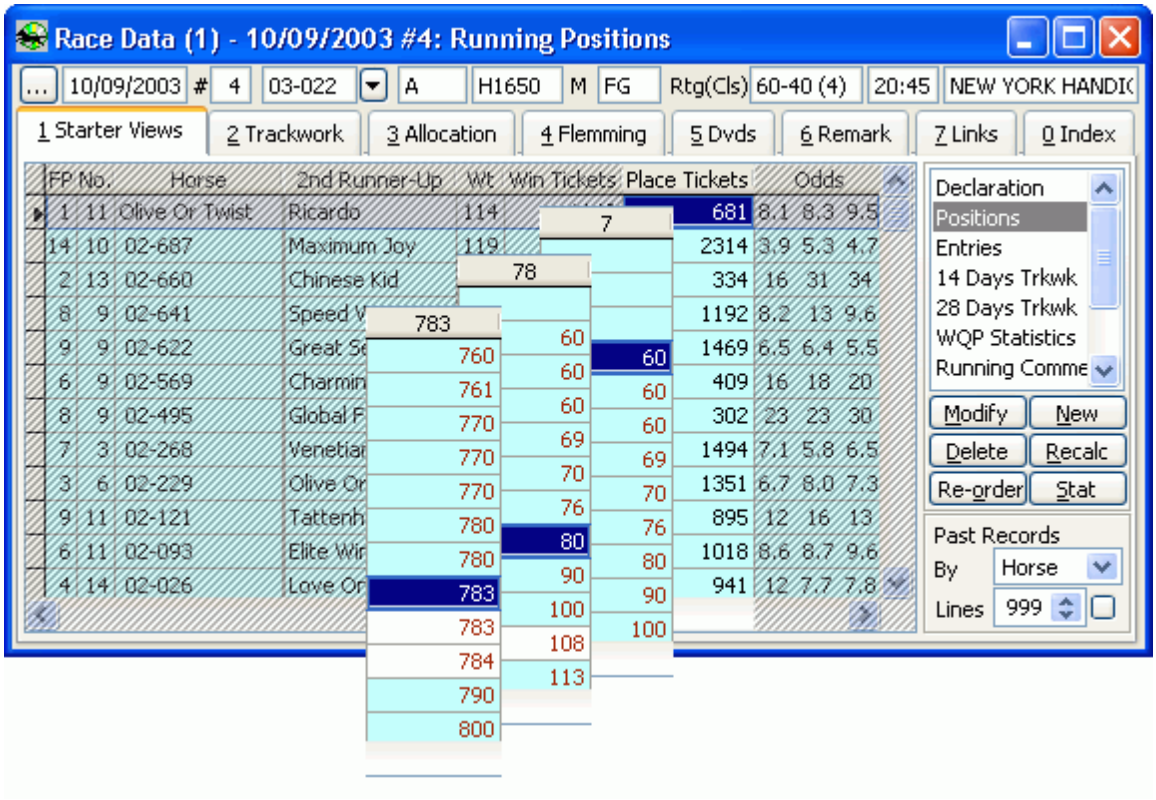
1. If you do an incremental search in a column that is not currently ordered, when you enter your first keystroke, the column will automatically be sorted into ascending order. However, if the column has already been sorted in descending order, RaceMate will not change its order.
2. No matter what you have keyed in, there will generally be 2 best matching records, one is larger than what you entered and the other is smaller than what you entered. RaceMate will always jump to the one at a lower position (visually).

### Incremental Number Search

An incremental search in a numeric column is similar to an incremental search in a text column except that the key strokes that you entered are displayed in the column heading. The figure below shows an incremental search for the value 783 in the Place Tickets field.

| Keystroke | Number Input | Nearest Number |
|---|---|---|
| 7 | 7 | 60 |
| 8 | 78 | 80 |
| 3 | 783 | 783 |

© Welkin Systems Limited

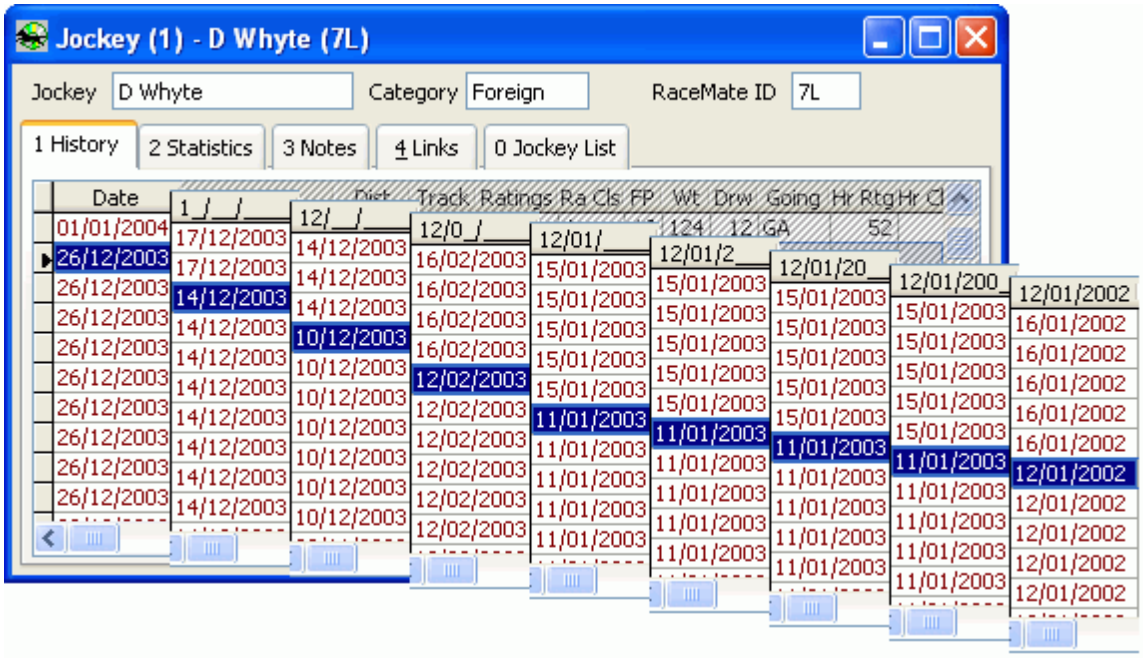

Race Data (1) - 10/09/2003 #4: Running Positions
10/09/2003 # 4 03-022 A H1650 M FG Rtg(Cls) 60-40 (4) 20:45 NEW YORK HANDIC
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP No. Horse 2nd Runner-Up Wt Win Tickets Place Tickets Odds
1 11 Olive Or Twist Ricardo 114 681 8.1 8.3 9.5
14 10 02-687 Maximum Joy 119 2314 3.9 5.3 4.7
2 13 02-660 Chinese Kid 334 16 31 34
8 9 02-641 Speed V 1192 8.2 13 9.6
9 9 02-622 Great Sc 1469 6.5 6.4 5.5
6 9 02-569 Charmin 409 16 18 20
8 9 02-495 Global F 302 23 23 30
7 3 02-268 Venetiar 1494 7.1 5.8 6.5
3 6 02-229 Olive Or 1351 6.7 8.0 7.3
9 11 02-121 Tatterh 895 12 16 13
6 11 02-093 Elite Wir 1018 8.6 8.7 9.6
4 14 02-026 Love Or 941 12 7.7 7.8
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
Modify New
Delete Recalc
Re-order Stat
Past Records
By Horse
Lines 999
7
78
783
760 761 770 770 770 780 780 783 783 784 790 800
60 60 60 69 70 76 80 90 100 108 113
60 60 60 69 70 76 80 90 100

**Incremental Date Search**

An incremental search in a date column is similar to an incremental search in a numeric column. The keystrokes that you entered are displayed in the column heading. In a date search, the cell that you start will be used as a template for the incremental search. Each keystroke that you key in will replace a character in the template. RaceMate will then use the latest template value to jump to the best matching record. The figure below shows an incremental search for the date 12/01/2002 in the Date column (already in descending order before searching). The value in cell (used as template) in which the search starts is 26/12/2003.

| Keystroke | Latest Template | Nearest Date |
|---|---|---|
| | 26/12/2003 | |
| 1 | 16/12/2003 | 14/12/2003 |
| 2 | 12/12/2003 | 10/12/2003 |
| 0 | 12/02/2003 | 12/02/2003 |
| 1 | 12/01/2003 | 11/01/2003 |
| 2 | 12/01/2003 | 11/01/2003 |
| 0 | 12/01/2003 | 11/01/2003 |
| 0 | 12/01/2003 | 11/01/2003 |
| 2 | 12/01/2002 | 12/01/2002 |

© Welkin Systems Limited




**Stop an Incremental Search**

You may press an arrow key or click on an area outside the highlighted cell to stop an incremental search.

## 4.5 Editing Data

**Edit Mode**

Some fields on RaceMate's forms are editable (e.g., the Note field of the Horse Form). However, by default, all the editable fields are locked to avoid accidental change in contents. You must first click on the [Edit] button of the toolbar (or press Ctrl+E) before editing the fields. After the [Edit] button is clicked, the editable fields on a form will change to light-green in color showing that the form is in Edit Mode. After editing is complete, you must click on the [Save] button to save the change.

**Enabling Direct Edit**

You may enable RaceMate's Direct Edit feature by selecting the [Direct Edit] option on the Options and System Information Form (in Tools menu). When Direct Edit mode is enabled, you can change data in editable fields directly without first clicking on the [Edit] button of the toolbar. When you press a key of the keyboard in an editable textbox, the form will automatically go into the Edit Mode. Textboxes will become light-green in color. You need to click on the [Save] button to save the change after editing is complete.

Please note that once you turn on the Direct Edit feature of RaceMate, you can no longer carry out incremental search in editable columns.

## 4.6 Handling Forms

Form is a window in RaceMate for displaying data. Basic techniques about using forms will be discussed in this section.

© Welkin Systems Limited

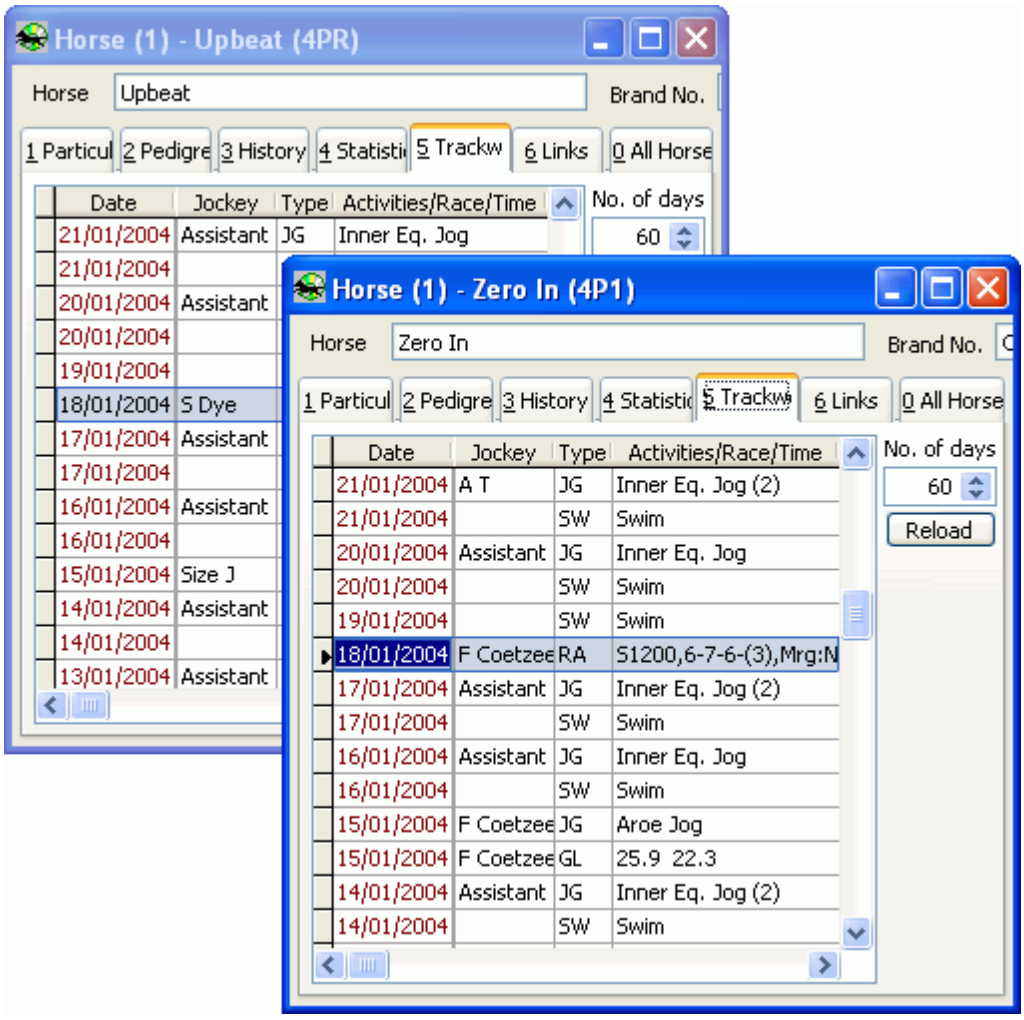
**>> Subtopics of this section <<<**

- Multiple Instances of a Form
- Freezing a Form
- Changing and Restoring Form Size

## 4.6.1 Multiple Instances of a Form

You may open a form several times for comparing different records of the same form. The figure below shows 2 instances of the Horse Form, one for "Upbeat" and the other for "Zero In". You may follow one of the ways below to open a new instance of a form:

1. Open the form again by choosing in RaceMate's main menu
2. Hold down the [Ctrl] key. Click on the form's button on RaceMate's Form Toolbar
3. Hold down the [Ctrl] key. Double-click on corresponding data column (for Horse, Jockey, Stable, Race) in a grid
4. Freeze an opened form. Double-click on corresponding data column (for Horse, Jockey, Stable, Race) in a grid




© Welkin Systems Limited

### 4.6.2 Freezing a Form

You may hyper-reference the data of a horse by double-clicking on a horse in a form. RaceMate will than activate a Horse form to display the record of the Horse. However, in order to conserve resources, RaceMate will try to try to minimize the number of active forms. If a Horse form has already been opened before your double-clicking, RaceMate will use that form to display the record rather than to open a new instance of the form.

Sometimes you may want to keep a particular record of a horse on the screen without being replaced by subsequent hyper-referencing. The solution is to freeze the form by:

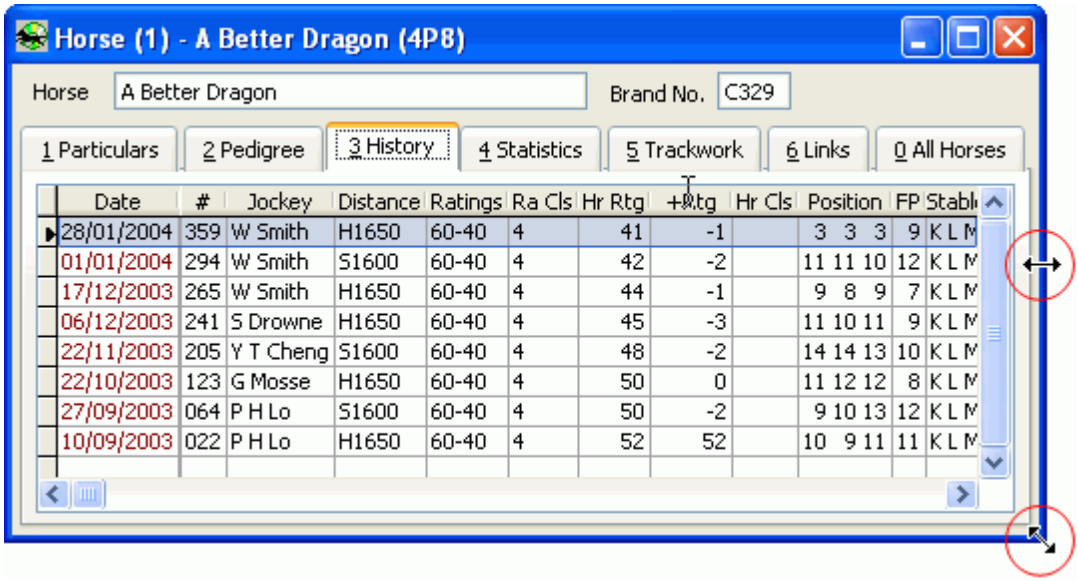
1. choosing [Freeze Form] from the [Record] menu, or
2. clicking on the [Freeze] button of the RaceMate toolbar

Such form freezing feature applies to all other data forms of RaceMate.

### 4.6.3 Changing and Restoring Form Size

**Resizing a Form**

Put the mouse pointer at a border (or the cross-section of 2 borders) of a form until it becomes a double arrow. Hold you the left mouse button and drag to resize the form.




**Restoring Original Form Size**

1. Double-click on a blank area of the form, or
2. Right-click on a blank area of the form and choose [Restore Form Size] from the shortcut menu.

## 4.7 Handling Grids

Many RaceMate data are display in grids of data forms. Necessary techniques of handling grids are discussed in this section.

© Welkin Systems Limited

**>> Subtopics of this section <<**

- Changing Column Width
- Changing Row Height
- Freezing Columns
- Changing a Column's Position
- Sorting Grid Columns
- Resetting Grid Layout

## 4.7.1 Changing Column Width

**Autofitting Width of a Column**

Double-click the area, between the column headers, to the immediate right of the column you want to resize.

**Autofitting Widths of All Columns**

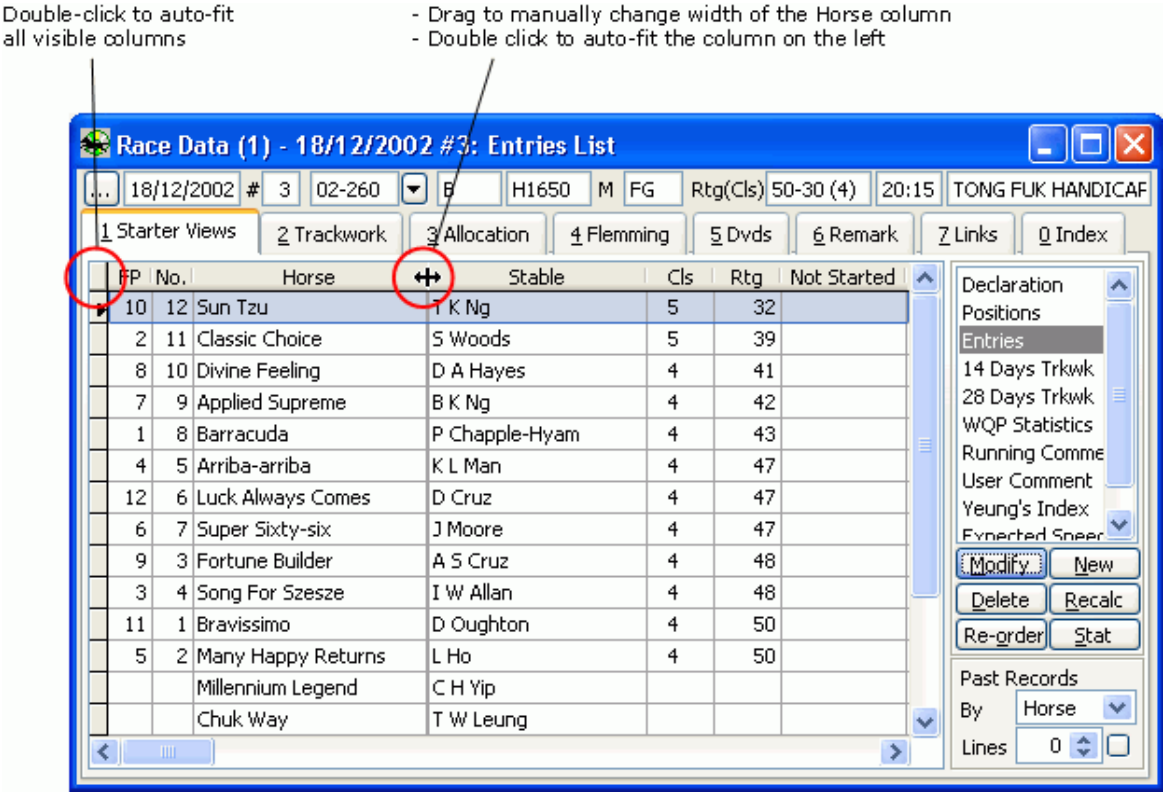
Double-click the square immediately preceding the first column header in the upper and leftmost corner of the grid.

**Manually Resizing a Column**

To change a column's width:

1. Put the mouse pointer on the right border of the column header until the pointer changes to
2. Hold down the mouse button and drag to left or right to adjust the column width.

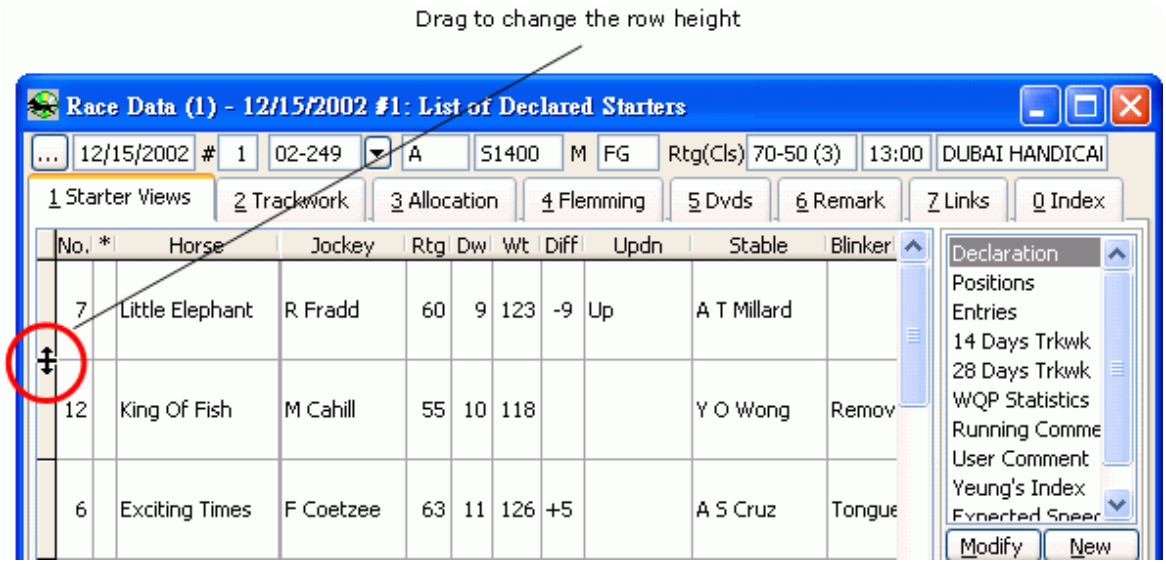
© Welkin Systems Limited


Double-click to auto-fit
all visible columns
- Drag to manually change width of the Horse column
- Double click to auto-fit the column on the left
Race Data (1) - 18/12/2002 #3: Entries List
18/12/2002 # 3 02-260 B H1650 M FG Rtg(Cls) 50-30 (4) 20:15 TONG FUK HANDICAF
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP No. Horse Stable Cls Rtg Not Started
10 12 Sun Tzu T K Ng 5 32
2 11 Classic Choice S Woods 5 39
8 10 Divine Feeling D A Hayes 4 41
7 9 Applied Supreme B K Ng 4 42
1 8 Barracuda P Chapple-Hyam 4 43
4 5 Arriba-arriba K L Man 4 47
12 6 Luck Always Comes D Cruz 4 47
6 7 Super Sixty-six J Moore 4 47
9 3 Fortune Builder A S Cruz 4 48
3 4 Song For Szesze I W Allan 4 48
11 1 Bravissimo D Oughton 4 50
5 2 Many Happy Returns L Ho 4 50
Millennium Legend C H Yip
Chuk Way T W Leung
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify
New
Delete
Recalc
Re-order
Stat
Past Records
By Horse
Lines 0

## 4.7.2 Changing Row Height

### Manually Changing Row Height

You can manually change the row height by moving the mouse pointer below the first row until the mouse pointer changes to ↕. Then keep holding down the mouse button and drag upwards or downwards to change the height.

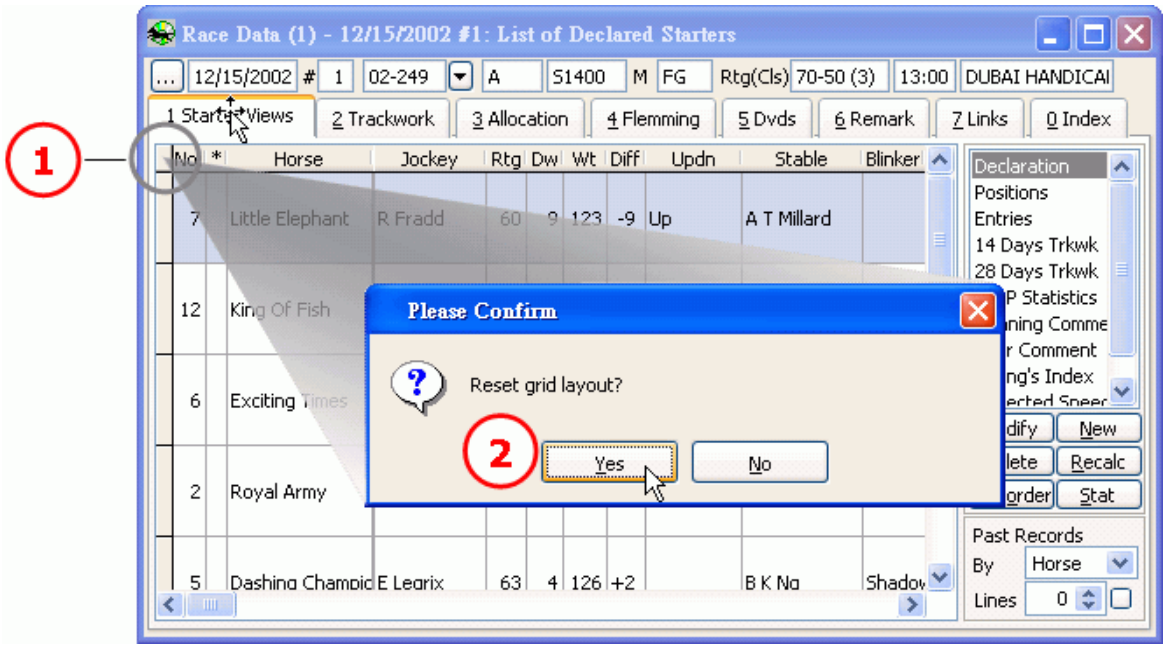

Drag to change the row height
Race Data (1) - 12/15/2002 #1: List of Declared Starters
12/15/2002 # 1 02-249 A S1400 M FG Rtg(Cls) 70-50 (3) 13:00 DUBAI HANDICAI
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
No. * Horse Jockey Rtg Dw Wt Diff Updn Stable Blinker
7 Little Elephant R Fradd 60 9 123 -9 Up A T Millard
12 King Of Fish M Cahill 55 10 118 Y O Wong Remov
6 Exciting Times F Coetzee 63 11 126 +5 A S Cruz Tongue
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify
New

### Restoring Original Row Height

If you want to restore to original row height, you can follow the steps below:

© Welkin Systems Limited

1. Click the square immediately preceding the first column header in the upper and leftmost corner of the grid.
2. When Please Confirm dialog box appears, click [Yes].




## 4.7.3 Freezing Columns

You can freeze columns in a data grid to keep certain columns visible while you scroll back and forth across the grid. Tabbing to the right does not cause locked columns to scroll out of the grid.

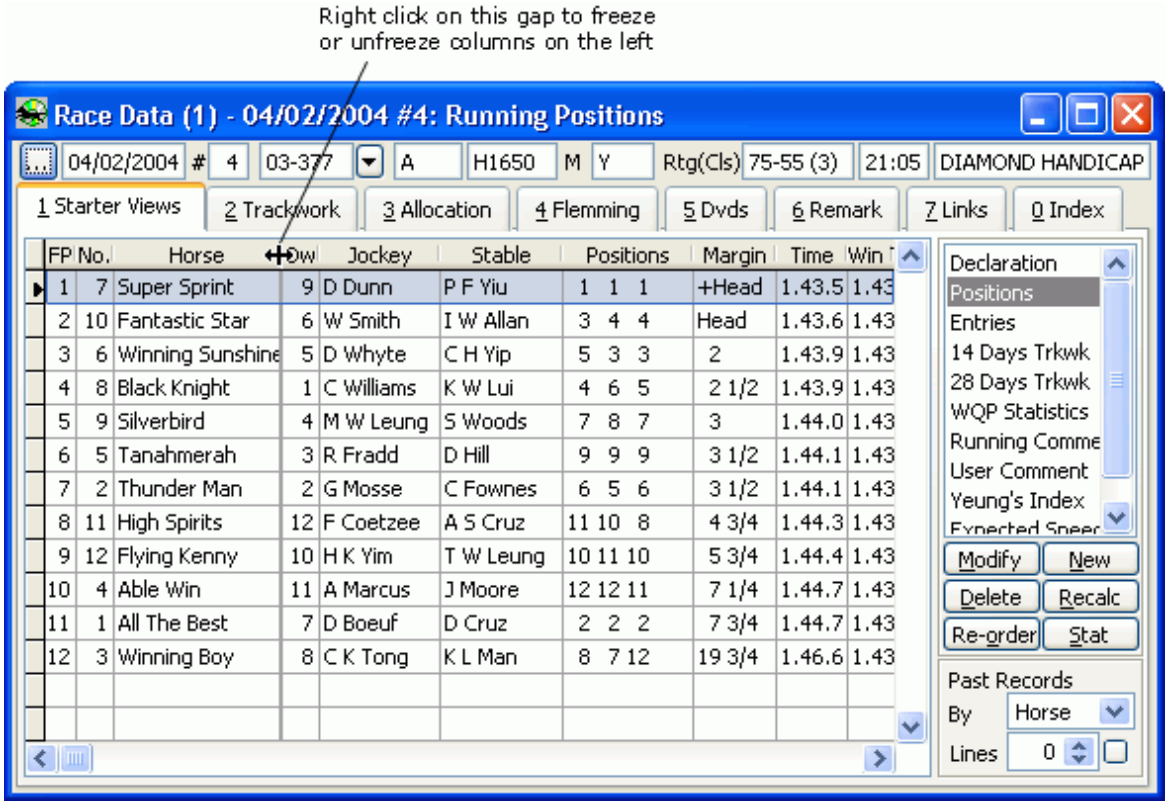
### Freezing Columns

To freeze columns, right-click the area, between the column headers, immediately following the rightmost column you want to lock. (If there are already some columns being frozen, you need to right-click twice. The first click unfreezes the existing locked columns and the actual freezing is done by the second click.)

### Unfreezing Columns

To unfreeze columns, right-click the area between the column headers.

© Welkin Systems Limited


Right click on this gap to freeze
or unfreeze columns on the left
Race Data (1) - 04/02/2004 #4: Running Positions
04/02/2004 # 4 03-377 A H1650 M Y Rtg(Cls) 75-55 (3) 21:05 DIAMOND HANDICAP
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP No. Horse Dw Jockey Stable Positions Margin Time Win
1 7 Super Sprint 9 D Dunn P F Yiu 1 1 1 +Head 1.43.5 1.43
2 10 Fantastic Star 6 W Smith I W Allan 3 4 4 Head 1.43.6 1.43
3 6 Winning Sunshine 5 D Whyte C H Yip 5 3 3 2 1.43.9 1.43
4 8 Black Knight 1 C Williams K W Lui 4 6 5 2 1/2 1.43.9 1.43
5 9 Silverbird 4 M W Leung S Woods 7 8 7 3 1.44.0 1.43
6 5 Tanahmerah 3 R Fradd D Hill 9 9 9 3 1/2 1.44.1 1.43
7 2 Thunder Man 2 G Mosse C Fownes 6 5 6 3 1/2 1.44.1 1.43
8 11 High Spirits 12 F Coetzee A S Cruz 11 10 8 4 3/4 1.44.3 1.43
9 12 Flying Kenny 10 H K Yim T W Leung 10 11 10 5 3/4 1.44.4 1.43
10 4 Able Win 11 A Marcus J Moore 12 12 11 7 1/4 1.44.7 1.43
11 1 All The Best 7 D Boeuf D Cruz 2 2 2 7 3/4 1.44.7 1.43
12 3 Winning Boy 8 C K Tong K L Man 8 7 12 19 3/4 1.46.6 1.43
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify New
Delete Recalc
Re-order Stat
Past Records
By Horse
Lines 0

## 4.7.4 Changing a Column's Position

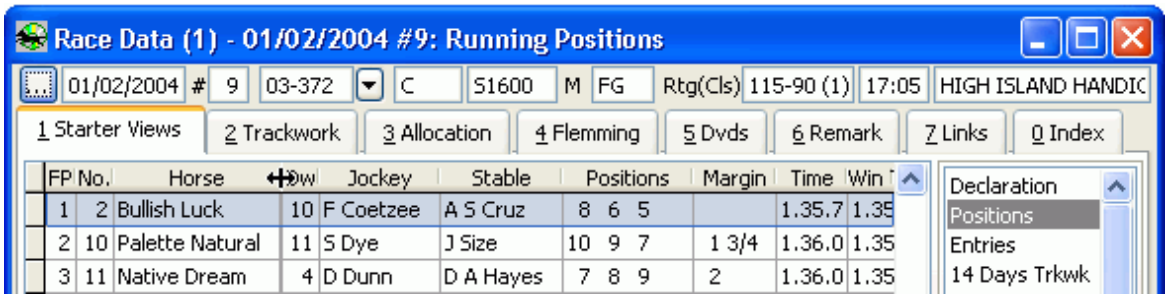
**To change the position of a column:**

1. Unfreeze leftmost columns, if any.
2. Put the mouse pointer on the column header until the pointer changes to ⬇
3. Hold down the mouse button and drag the column to the desired position.

**Example: Moving the Stable column to the right of the Horse column.**

1. Unfreeze leftmost columns

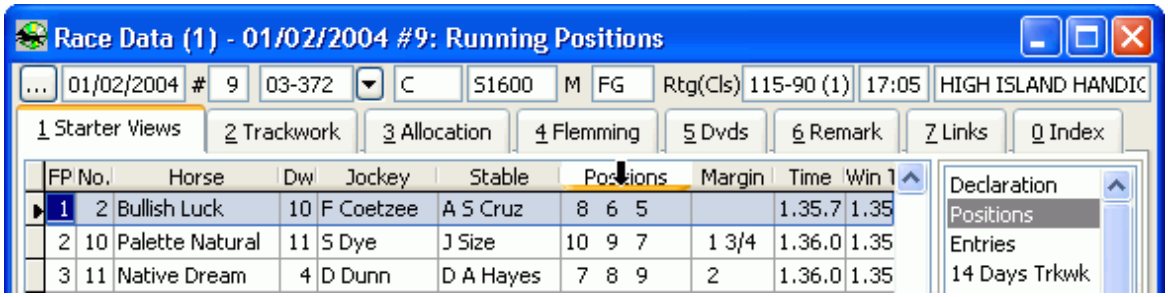
(If there is no frozen column at the left, please skip this step)
Right click at the gap between the headings of the columns Horse and Dw.


Race Data (1) - 01/02/2004 #9: Running Positions
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP No. Horse Dw Jockey Stable Positions Margin Time Win
1 2 Bullish Luck 10 F Coetzee A S Cruz 8 6 5 1.35.7 1.35
2 10 Palette Natural 11 S Dye J Size 10 9 7 1 3/4 1.36.0 1.35
3 11 Native Dream 4 D Dunn D A Hayes 7 8 9 2 1.36.0 1.35
Declaration
Positions
Entries
14 Days Trkwk

2. Move Column

Move the mouse point to the column heading of the Position column.

© Welkin Systems Limited


Race Data (1) - 01/02/2004 #9: Running Positions
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Remark 7 Links 0 Index
FP No. Horse Dw Jockey Stable Positions Margin Time Win 1
1 2 Bullish Luck 10 F Coetzee A S Cruz 8 6 5 1.35.7 1.35
2 10 Palette Natural 11 S Dye J Size 10 9 7 1 3/4 1.36.0 1.35
3 11 Native Dream 4 D Dunn D A Hayes 7 8 9 2 1.36.0 1.35
Declaration
Positions
Entries
14 Days Trkwk

Drag the Position column to the right of the Horse column.

Race Data (1) - 01/02/2004 #9: Running Positions
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Remark 7 Links 0 Index
FP No. Horse Positions Dw Jockey Stable Margin Time Win 1
Declaration
Positions
Entries
14 Days Trkwk

Release the mouse button.

Race Data (1) - 01/02/2004 #9: Running Positions
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Remark 7 Links 0 Index
FP No. Horse Positions Dw Jockey Stable Margin Time Win 1
1 2 Bullish Luck 8 6 5 10 F Coetzee A S Cruz 1.35.7 1.35
2 10 Palette Natural 10 9 7 11 S Dye J Size 1 3/4 1.36.0 1.35
3 11 Native Dream 7 8 9 4 D Dunn D A Hayes 2 1.36.0 1.35
Declaration
Positions
Entries
14 Days Trkwk

3. Restore Column Freezing

Right click at the gap between the headings of the columns Horse and Dw to restore the frozen status of the 3 leftmost columns.

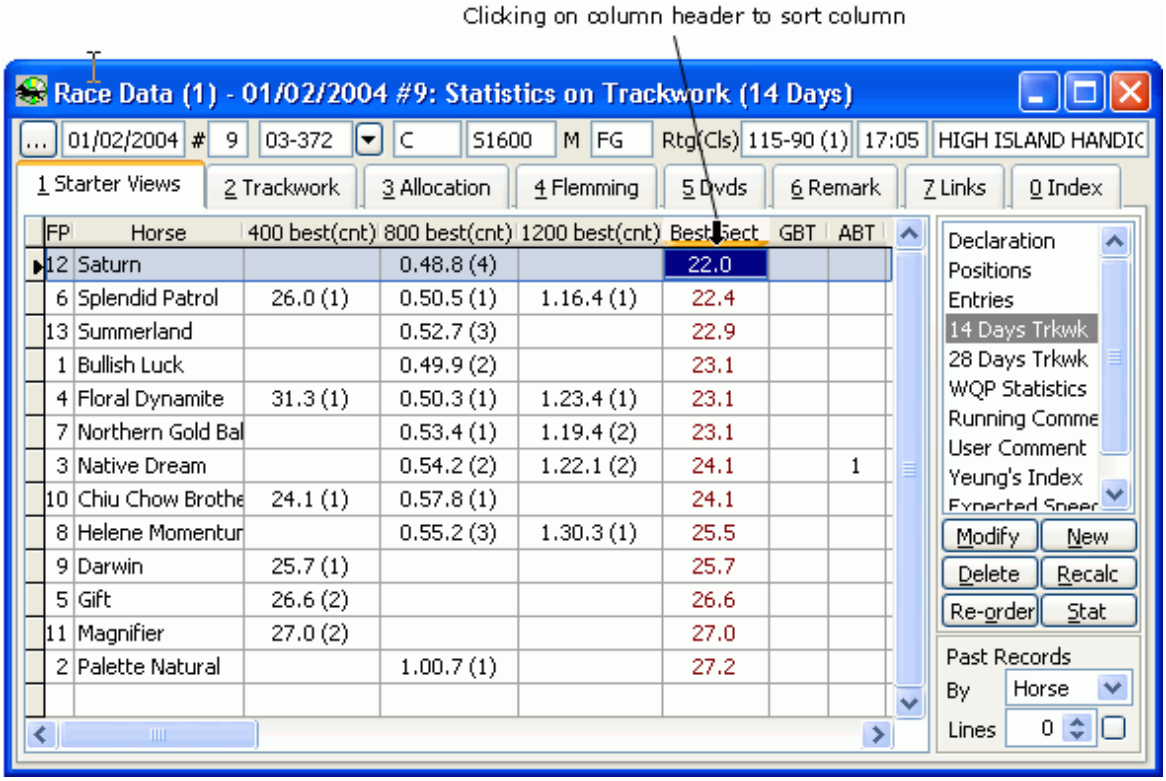
Race Data (1) - 01/02/2004 #9: Running Positions
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Remark 7 Links 0 Index
FP No. Horse Positions Dw Jockey Stable Margin Time Win 1
1 2 Bullish Luck 8 6 5 10 F Coetzee A S Cruz 1.35.7 1.35
2 10 Palette Natural 10 9 7 11 S Dye J Size 1 3/4 1.36.0 1.35
3 11 Native Dream 7 8 9 4 D Dunn D A Hayes 2 1.36.0 1.35
4 12 Floral Dynamite 4 5 6 3 D Boeuf K W Lui 2 1/2 1.36.1 1.35
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk

## 4.7.5 Sorting Grid Columns

### Single Column Sort

You can sort data in a column by clicking on the column header. On each clicking, the data will be sorted in ascending or descending order alternately.

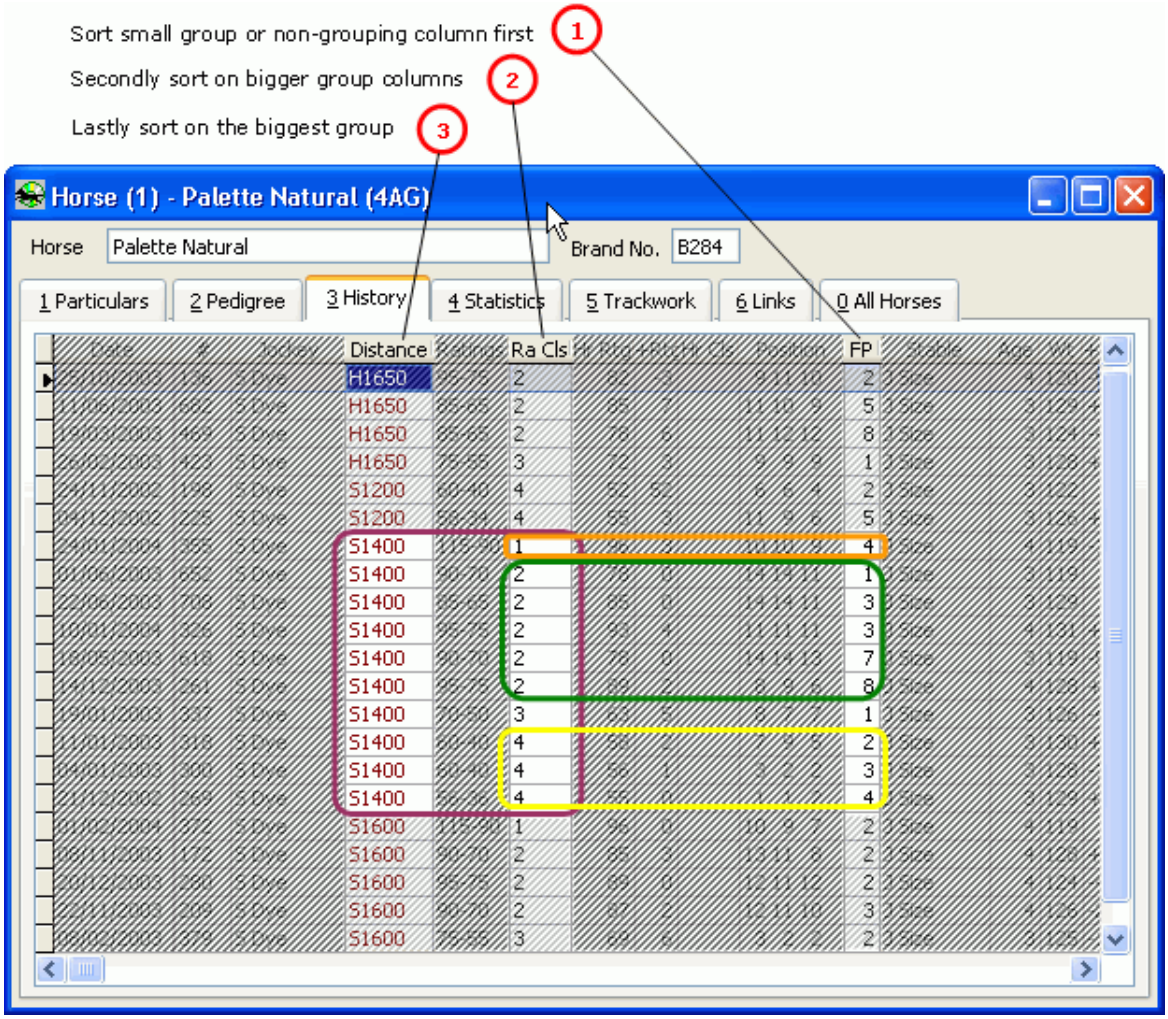
© Welkin Systems Limited


Clicking on column header to sort column
Race Data (1) - 01/02/2004 #9: Statistics on Trackwork (14 Days)
01/02/2004 # 9 03-372 C S1600 M FG Rtg(Cls) 115-90 (1) 17:05 HIGH ISLAND HANDIC
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP Horse 400 best(cnt) 800 best(cnt) 1200 best(cnt) Best Sect GBT ABT
12 Saturn 0.48.8 (4) 22.0
6 Splendid Patrol 26.0 (1) 0.50.5 (1) 1.16.4 (1) 22.4
13 Summerland 0.52.7 (3) 22.9
1 Bullish Luck 0.49.9 (2) 23.1
4 Floral Dynamite 31.3 (1) 0.50.3 (1) 1.23.4 (1) 23.1
7 Northern Gold Bal 0.53.4 (1) 1.19.4 (2) 23.1
3 Native Dream 0.54.2 (2) 1.22.1 (2) 24.1 1
10 Chiu Chow Brothe 24.1 (1) 0.57.8 (1) 24.1
8 Helene Momentur 0.55.2 (3) 1.30.3 (1) 25.5
9 Darwin 25.7 (1) 25.7
5 Gift 26.6 (2) 26.6
11 Magnifier 27.0 (2) 27.0
2 Palette Natural 1.00.7 (1) 27.2
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify
New
Delete
Recalc
Re-order
Stat
Past Records
By Horse
Lines 0

### Multicolumn Sort

You may sort more than one column simultaneously. The basic rule of doing a multicolumn sort is: "small groups first, large groups last". For example, say, we want to sort the finishing position of a horse for each Distance - Race Class combination. We will order distance so that the same distance goes into a group (largest group, do last). For starts of the same distance, records are group be race classes. Within the same race class records will be ordered by finishing positions (smallest group, do first).

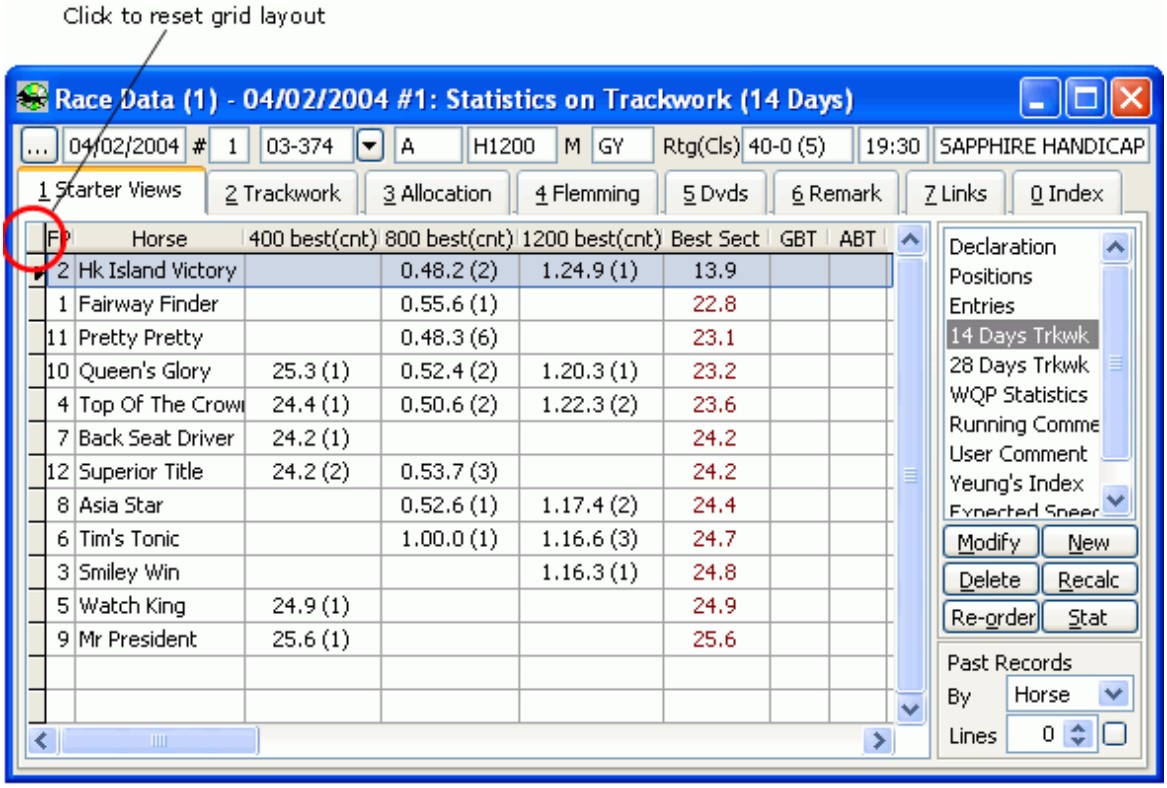
© Welkin Systems Limited




## 4.7.6 Resetting Grid Layout

Changes to grid layouts, including sorting, column moving, column resizing, can be reset by clicking on the left top rectangle of the grid. The system will ask you to confirm if you are going to reset the grid layout. Answer "Yes" to reset.

© Welkin Systems Limited

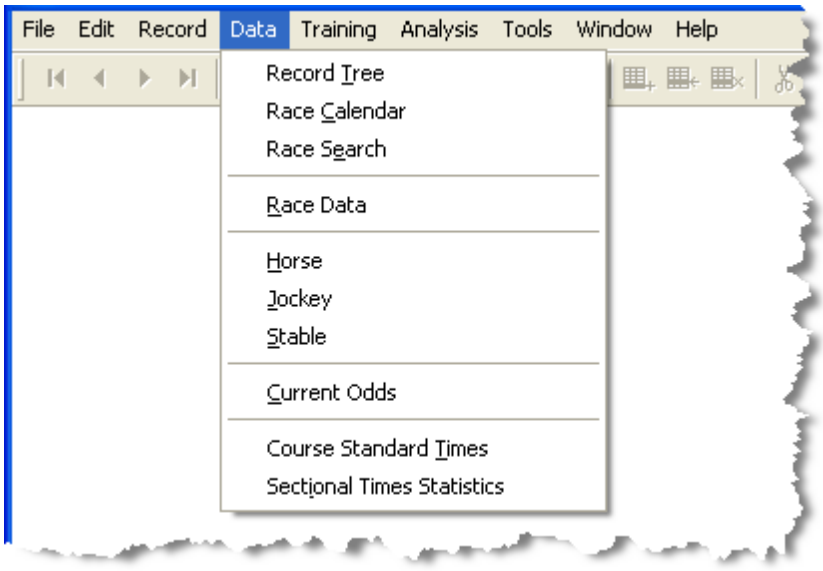

Click to reset grid layout
Race Data (1) - 04/02/2004 #1: Statistics on Trackwork (14 Days)
04/02/2004 # 1 03-374 A H1200 M GY Rtg(Cls) 40-0 (5) 19:30 SAPPHIRE HANDICAP
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
F# Horse 400 best(cnt) 800 best(cnt) 1200 best(cnt) Best Sect GBT ABT
2 Hk Island Victory 0.48.2 (2) 1.24.9 (1) 13.9
1 Fairway Finder 0.55.6 (1) 22.8
11 Pretty Pretty 0.48.3 (6) 23.1
10 Queen's Glory 25.3 (1) 0.52.4 (2) 1.20.3 (1) 23.2
4 Top Of The Crow 24.4 (1) 0.50.6 (2) 1.22.3 (2) 23.6
7 Back Seat Driver 24.2 (1) 24.2
12 Superior Title 24.2 (2) 0.53.7 (3) 24.2
8 Asia Star 0.52.6 (1) 1.17.4 (2) 24.4
6 Tim's Tonic 1.00.0 (1) 1.16.6 (3) 24.7
3 Smiley Win 1.16.3 (1) 24.8
5 Watch King 24.9 (1) 24.9
9 Mr President 25.6 (1) 25.6
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify
New
Delete
Recalc
Re-order
Stat
Past Records
By Horse
Lines 0

© Welkin Systems Limited

# Part V

# 5 Main Data Forms

You may pull down the Data menu of RaceMate to get access to RaceMate's common data forms including Race Calendar, Race Data, Horse, Jockey, and Stable. The Race Calendar form is the hub of race data. Very often you start your data search from the Race Calendar and chain to other data forms from via the hyper-linking feature of RaceMate.




**>>> Subtopics of this section <<<**

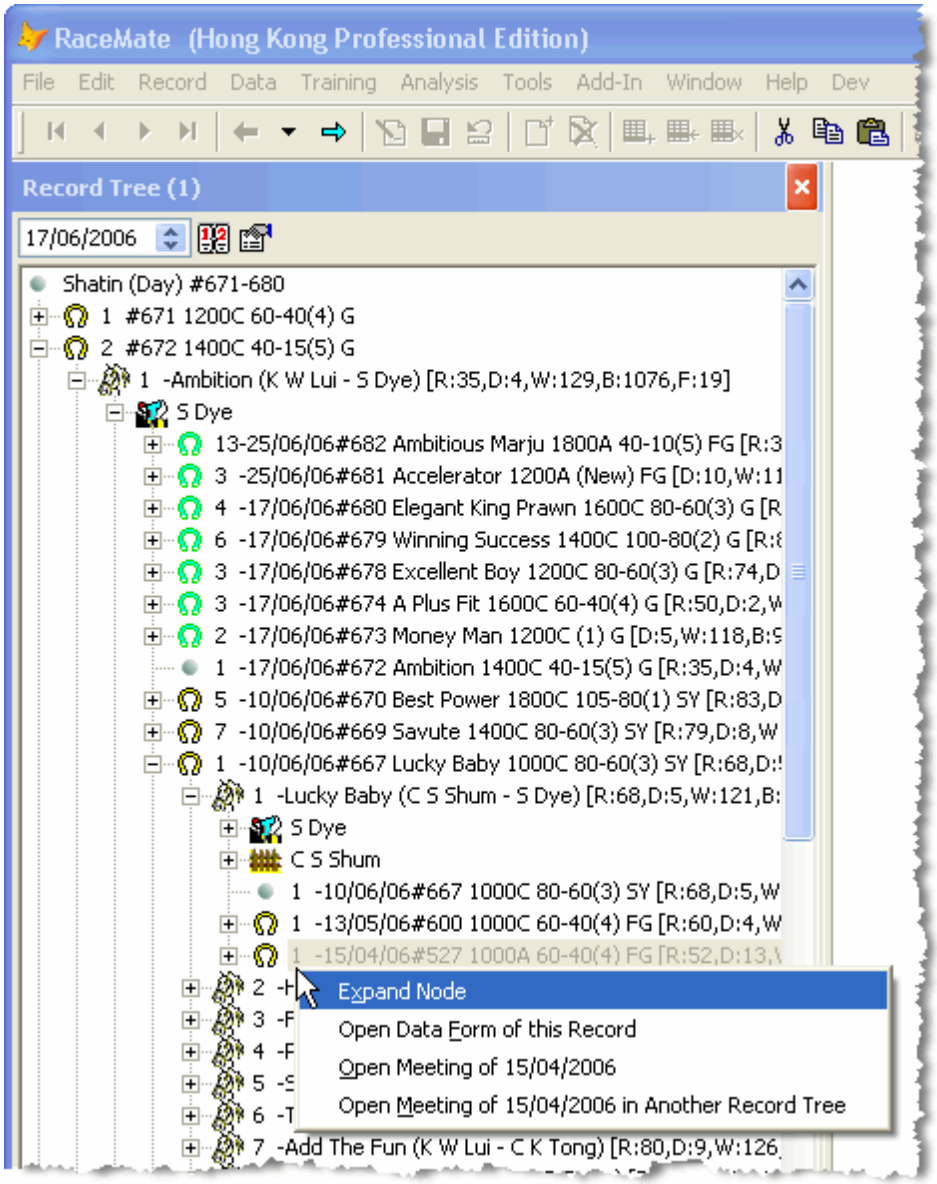
- Record Tree Form
- Race Calendar Form
- Race Data Form
- Race Search Form
- Jockey Silks
- Horse Form
- Jockey Form
- Stable Form
- Current Odds from Jockey Club
- Daily Track Work and Barrier Trial
- Jump by Double-Clicking a Column
- Comprehensive Statistics and Record Query Form

## 5.1 Record Tree Form

The Record Tree form combines features of the Calendar, Race, Horse, Jockey and Stable forms into an expandable and collapsible data tree. Both backward (past) and forward (future) records can be shown in the tree. You can follow detailed history of each horse, jockey and stable by expanding the nodes (unlimited levels) of the tree.

The Record Tree is actually a data form navigator. You can opening a corresponding data form by clicking on a record on the tree. For example, clicking on a race record will open the Race Data form while clicking on a horse record will open the Horse Data form, and so on.

© Welkin Systems Limited


RaceMate (Hong Kong Professional Edition)
File Edit Record Data Training Analysis Tools Add-In Window Help Dev
Record Tree (1)
17/06/2006
Shatin (Day) #671-680
1 #671 1200C 60-40(4) G
2 #672 1400C 40-15(5) G
1 -Ambition (K W Lui - S Dye) [R:35,D:4,W:129,B:1076,F:19]
S Dye
13-25/06/06#682 Ambitious Marju 1800A 40-10(5) FG [R:3
3 -25/06/06#681 Accelerator 1200A (New) FG [D:10,W:11
4 -17/06/06#680 Elegant King Prawn 1600C 80-60(3) G [R
6 -17/06/06#679 Winning Success 1400C 100-80(2) G [R:
3 -17/06/06#678 Excellent Boy 1200C 80-60(3) G [R:74,D
3 -17/06/06#674 A Plus Fit 1600C 60-40(4) G [R:50,D:2,W
2 -17/06/06#673 Money Man 1200C (1) G [D:5,W:118,B:9
1 -17/06/06#672 Ambition 1400C 40-15(5) G [R:35,D:4,W
5 -10/06/06#670 Best Power 1800C 105-80(1) SY [R:83,D
7 -10/06/06#669 Savute 1400C 80-60(3) SY [R:79,D:8,W
1 -10/06/06#667 Lucky Baby 1000C 80-60(3) SY [R:68,D:
1 -Lucky Baby (C S Shum - S Dye) [R:68,D:5,W:121,B:
S Dye
C S Shum
1 -10/06/06#667 1000C 80-60(3) SY [R:68,D:5,W
1 -13/05/06#600 1000C 60-40(4) FG [R:60,D:4,W
Expand Node
Open Data Form of this Record
Open Meeting of 15/04/2006
Open Meeting of 15/04/2006 in Another Record Tree
7 -Add The Fun (K W Lui - C K Tong) [R:80,D:9,W:126

### Changing Meeting Date

A record tree displays record information about a meeting. The meeting can be selected in one of the following ways:

1. Enter the a date directly into the left top date box of the Record Tree form
2. Clicking on the up or down arrow (or push the Up and Down arrows on the keyboard) in the date box to select a date.
3. Click on the [Race Calendar] button on the top bar of the record tree to activate the Race Calendar. While you choose a date on the Race Calendar, the meeting date of Record Tree will automatically be synchronized.
4. Right-click on a tree node and choose [Open Meeting of ...] on the quick menu to set the meeting date to the race date of the selected node or choose [Open Meeting of ... in Another Record Tree] to open the meeting in another Record Tree.

### Icons on Record Tree

© Welkin Systems Limited

| Icon | Description |
|---|---|
| Yellow horseshoe | Race on or before the underlying race meeting date (past record) |
| Green horseshoe | Race after the underlying race meeting date (future record) |
| Grey dot at tree root | Meeting information |
| Grey dot between green and yellow horseshoe icons | The race in which the horse (or jockey or stable) is currently running. |
| Horse | Horse record |
| Jockey | Jockey record |
| Stable | Stable record |

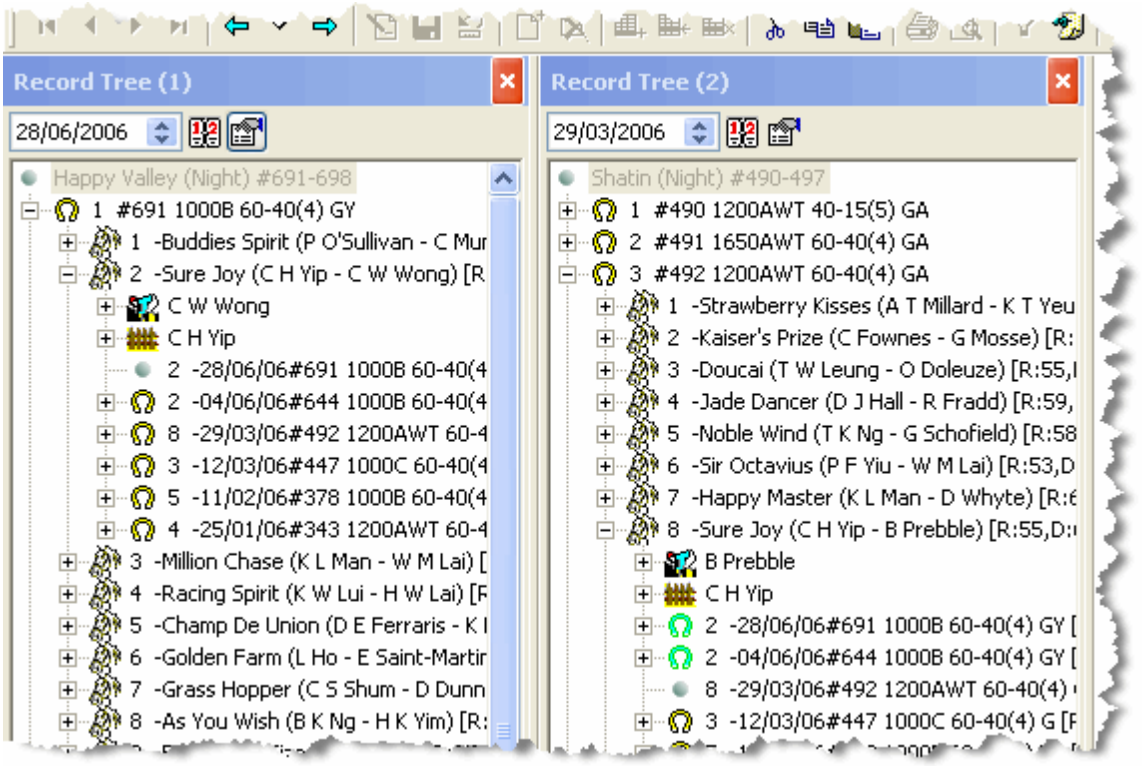
**Simultaneously Open Two or More Record Trees**

You may open two or more Record Tree forms simultaneous to compare two different sets of records. You may open another copy of Record Tree in one of the following ways:

1. Choose [Record Tree] on the [Data] menu of the RaceMate application menu.
2. Right-click on a tree node and choose [Open Meeting of ... in Another Record Tree] on the quick menu to set the meeting date to the race date of the selected node.

If you follow the 2nd way above to open a meeting in another record tree, RaceMate will try to find another currently open Record Tree form (other than the one you are working in). If one is found, the tree for the meeting will be open in this form. If there is no other currently open Record Tree form, RaceMate will then open a new one. If you want RaceMate to a new one instead of using an existing one, hold down the [Ctrl] button before you choose the [Open Meeting of ... in Another Record Tree] on the quick menu.

© Welkin Systems Limited

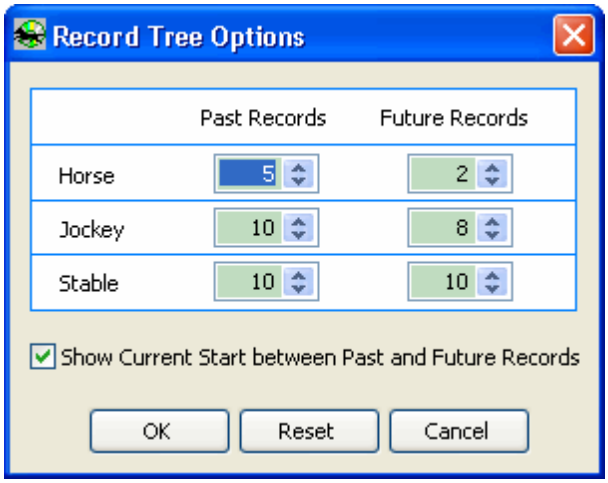

Record Tree (1)
28/06/2006
Happy Valley (Night) #691-698
1 #691 1000B 60-40(4) GY
1 -Buddies Spirit (P O'Sullivan - C Mur
2 -Sure Joy (C H Yip - C W Wong) [R
C W Wong
C H Yip
2 -28/06/06#691 1000B 60-40(4
2 -04/06/06#644 1000B 60-40(4
8 -29/03/06#492 1200AWT 60-4
3 -12/03/06#447 1000C 60-40(4
5 -11/02/06#378 1000B 60-40(4
4 -25/01/06#343 1200AWT 60-4
3 -Million Chase (K L Man - W M Lai) [
4 -Racing Spirit (K W Lui - H W Lai) [R
5 -Champ De Union (D E Ferraris - K I
6 -Golden Farm (L Ho - E Saint-Martir
7 -Grass Hopper (C S Shum - D Dunn
8 -As You Wish (B K Ng - H K Yim) [R:
Record Tree (2)
29/03/2006
Shatin (Night) #490-497
1 #490 1200AWT 40-15(5) GA
2 #491 1650AWT 60-40(4) GA
3 #492 1200AWT 60-40(4) GA
1 -Strawberry Kisses (A T Millard - K T Yeu
2 -Kaiser's Prize (C Fownes - G Mosse) [R:
3 -Doucai (T W Leung - O Doleuze) [R:55,I
4 -Jade Dancer (D J Hall - R Fradd) [R:59,
5 -Noble Wind (T K Ng - G Schofield) [R:58
6 -Sir Octavius (P F Yiu - W M Lai) [R:53,D
7 -Happy Master (K L Man - D Whyte) [R:6
8 -Sure Joy (C H Yip - B Prebble) [R:55,D:
B Prebble
C H Yip
2 -28/06/06#691 1000B 60-40(4) GY [
2 -04/06/06#644 1000B 60-40(4) GY [
8 -29/03/06#492 1200AWT 60-40(4)
3 -12/03/06#447 1000C 60-40(4) G [F

### Placing Record Tree on Another Monitor

A Record Tree is by default a dockable form. You may hold down your mouse on the title bar of a Record Tree and drag it out of the main RaceMate window or even put it into another monitor (if your computer has more than one monitor installed). Putting the Record Tree into another monitor can maximize your working space.

### Configuring the Record Tree

Click on the Option button at the top of the Record Tree to open the Record Tree Options form. On the form, you may specify numbers of past records and future records to be displayed on the tree. If you want to hide the current start record (grey dot between green and yellow horseshoes), you may uncheck the [Show Current Start between Past and Future Records] checkbox.

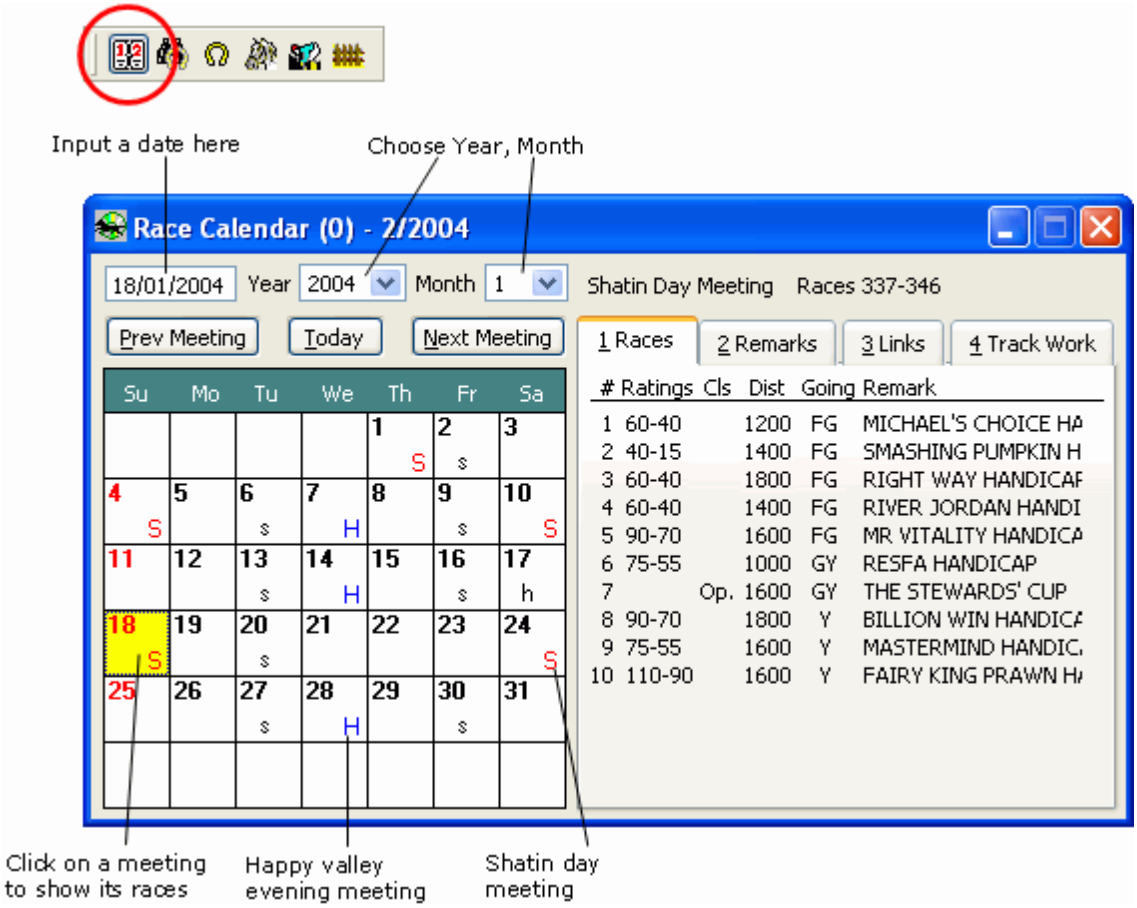



© Welkin Systems Limited

## 5.2 Race Calendar Form

When RaceMate starts, the Race Calendar will automatically be shown. To manually open the Race Calendar, you may

1. click on the [Calendar] button of the RaceMate toolbar, or
2. select [Race Calendar] from the data menu


Input a date here
Choose Year, Month
Race Calendar (0) - 2/2004
18/01/2004
Year 2004
Month 1
Shatin Day Meeting Races 337-346
Prev Meeting
Today
Next Meeting
1 Races
2 Remarks
3 Links
4 Track Work
# Ratings Cls Dist Going Remark
1 60-40 1200 FG MICHAEL'S CHOICE HA
2 40-15 1400 FG SMASHING PUMPKIN H
3 60-40 1800 FG RIGHT WAY HANDICAF
4 60-40 1400 FG RIVER JORDAN HANDI
5 90-70 1600 FG MR VITALITY HANDICA
6 75-55 1000 GY RESFA HANDICAP
7 Op. 1600 GY THE STEWARDS' CUP
8 90-70 1800 Y BILLION WIN HANDICA
9 75-55 1600 Y MASTERMIND HANDIC.
10 110-90 1600 Y FAIRY KING PRAWN HA
Su Mo Tu We Th Fr Sa
Click on a meeting to show its races
Happy valley evening meeting
Shatin day meeting

### Calendar Pane

On the left of the calendar form, a calendar marked with race meetings is show. You may jump to a calendar day via one of the following ways:

- Show a month by using the Month and Day dropdowns at the top of the form. Then click on a day
- Click on the [Previous Meeting], [Next Meetings], and [Today] buttons
- Enter directly a date into the left-top date box

After you have selected a day, the information about the day will be show on the pages on the right-hand side of the form. You will see the following symbols on the calendar:

| Symbol | HK Editions | |
|---|---|---|
| H | Happy Valley meeting | Red for day meetings, Blue for evening meetings |
| S | Shatin meeting | |
| D | / | |

© Welkin Systems Limited

| | |
|---|---|
| N | / |
| h | Happy Valley Barrier Trials |
| s | Shatin Barrier Trials |
| * | / |

**Races Page**

The first page on the right side lists races of the day highlighted on the calendar. You may click on a race to open the Race Data form to show the data about the race. If there is no meeting on the day, the page will show a message "No Race Today".

**Remarks Page**

The Remarks page shows the remarks from the Jockey Club for the day, if any. You may enter your own remarks in the UserRemarks edit box on the page.

**Links Page**

The Links page shows a set of autolinks for opening the multimedia resources, including videos and photos, of the Jockey Clubs. You may add your own multimedia resource links into the page as well. (See Autolinks of the Multimedia Resources section for details)

**Track Work Page**

You may click on the [List Track Work Horses] button at the bottom of the page to show all track work records of the day. The Track Work page also shows the barrier trials of the day, if any. You may click on a barrier trial to view the details about the trial on the Barrier Trial Form.

## 5.3 Race Data Form

The "Race Data" form of RaceMate displays data about races. It is can be opened by:

- clicking on a race inside the Race Calendar
- selecting [Race Data] from the [Data] menu

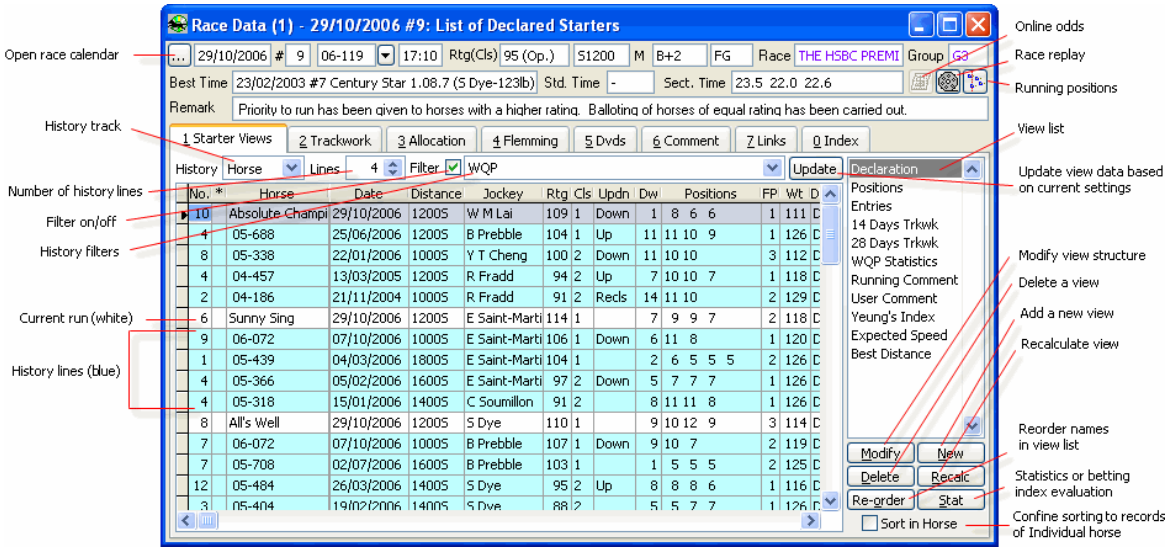
© Welkin Systems Limited

**>>> Subtopics of this section <<<**

- Starter View Page (Race Data Form)
- Trackwork Page (Race Data Form)
- Allocation Page (Race Data Form)
- Flemming Page (Race Data Form)
- Dividends Page (Race Data Form)
- Remark Page (Race Data Form)
- Links Page (Race Data Form)
- Index Page (Race Data Form)
- Print Race Data (Race Data Form)

## 5.3.1 Starter View Page (Race Data Form)

Starter Views is the first page of the Race Data Form. At the top of the Race Data form, there are a couple of lines showing some basic information of the race. Click on the leftmost button on this line to open the Race Calendar. Click on the small button with a down arrow to open the Race Search form.




### List Starter Views

You may select a view by clicking on the view list on the right of the page. RaceMate predefines a set of built-in views to present different information including statistics. You may add your own views by clicking on the new button. There is no limit on the number of user-defined views.

### Showing Past Records

Select a "history track" in the "History" dropdown list at the top of the Starter Views page. Select number of past record lines by changing the "Lines" number in the spinner next to the dropdown. You may optionally set a filter by pulling down the filter list next to the spinner. Then click on the [Update] button show past records in the view. The following past record tracks are available:

- Horse: show past records of each horse

© Welkin Systems Limited

- Jockey: show past records of each jockey
- Stable: show past records of each stable

**Filter On/Off**

The filter set by using the filter dropdown list can temporarily turn off by unchecking the [Filter] checkbox and later turn it on again by checking the box again.

**Useful Shortcuts**

- 0/999 Toggle: Double-click on the "Past Records Lines" spinner to toggle number of history lines between 0 and 999
- 0/n Toggle: Double-click on the [Lines] label on the left of the spinner to toggle between showing past records and hiding past records. Say, you have set the number of past record lines to 5 and want to temporarily hide the past record lines. You can simply double-click on the [Lines] label to hide all past records. Later when you want to show the 5 history lines again, just double-click again on the label, the 5 history lines will be shown again.

**Sorting**

You can order rows of a view based on the data of a column by clicking on the column header. Click once to do an ascending sort. Click once again to change it to descending order. Normally sorting is done over all history rows. If you want to sort records of individual horses, you may check the [Sort in Horse] checkbox.

**Recalculating View**

Click on the [Recalc] button to recalculate and update the data of a view. However filters will not be reevaluated. If you want to filter the records again, you may hold down the [Shift] key before clicking on the [Recalc] button.

Statistical Views and Betting Index

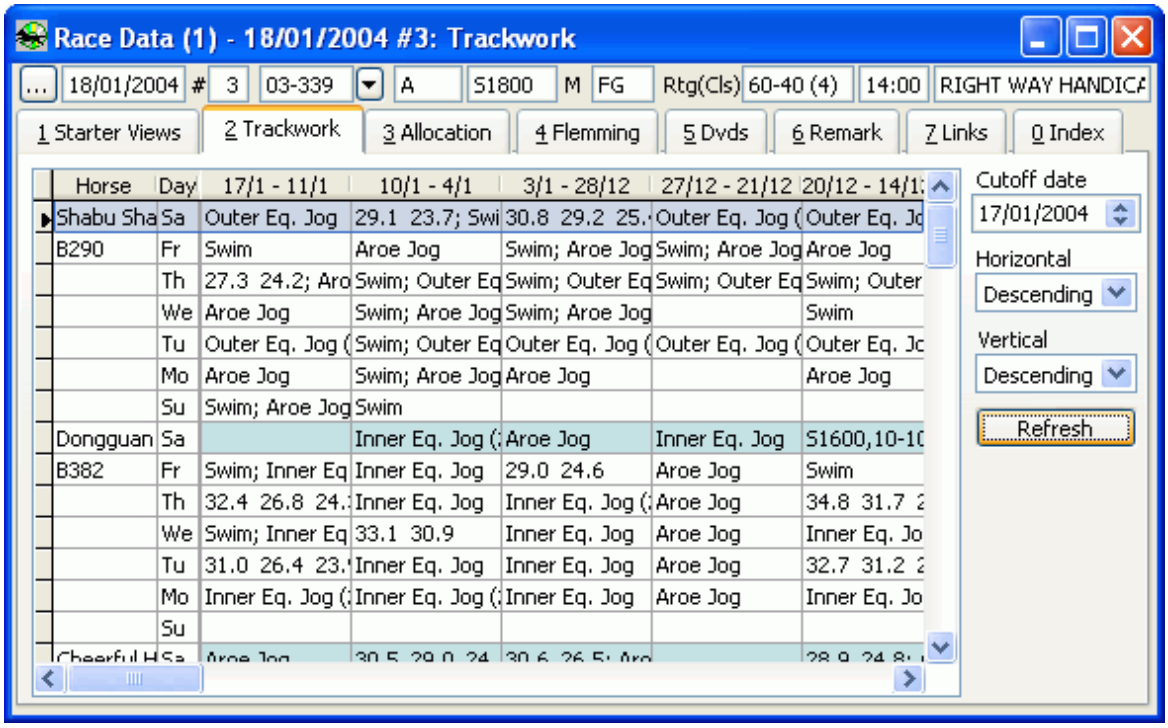
There are two special types of views which will make use of the [Stat] button.

- Statistical Views: Views created purely for doing statistics are called "statistical views" or statistical model. Statistical results of a statistical view are shown by clicking on the [Stat] button.
- Betting Index Views: A betting index view consists of a betting index column, which is a calculated column based on values of other columns of the same view. Betting indexes are constructed to predict the winning chance of starters. The validity or predictability of betting indexes can be evaluated by clicking on the [Stat] button.

### 5.3.2 Trackwork Page (Race Data Form)

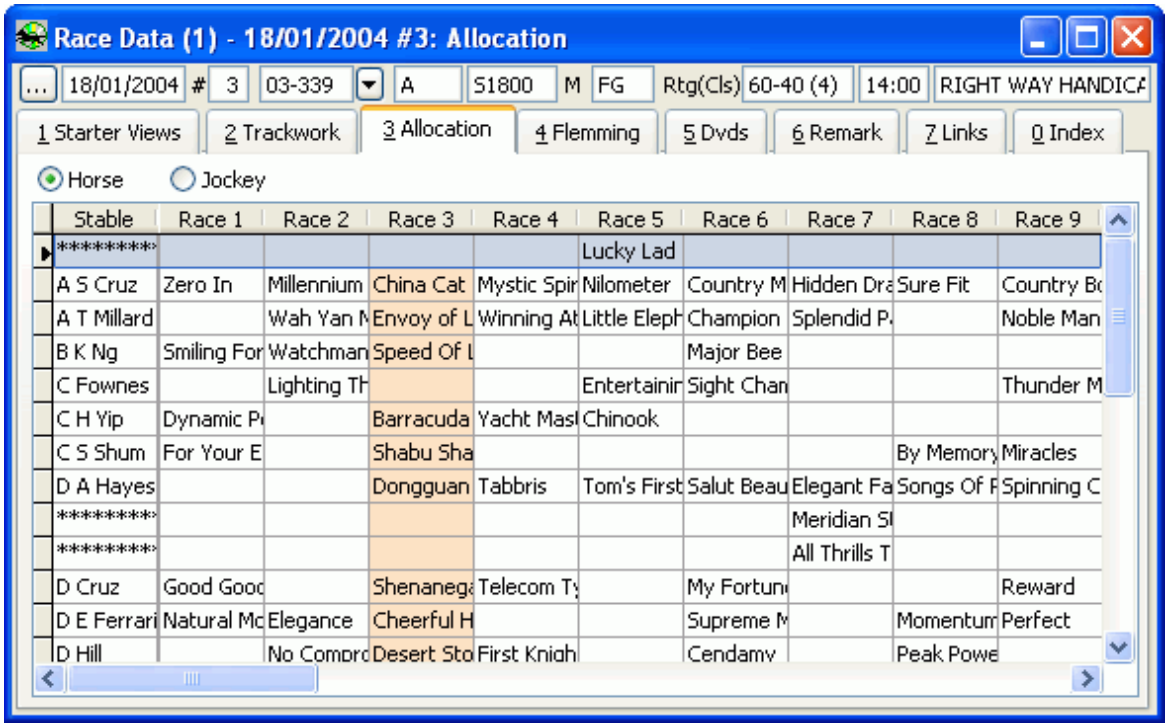
The Trackwork page of the Race Data Form shows track work of the starter during the latest 7 weeks. The nearest week will be put into the leftmost column and the nearest day in a week will be put in at the top. You may change such default order in the [Horizontal] and [Vertical] dropdowns. You may also display track work of earlier days by changing the [Curoff date] value. After you have made any change, you must click on the [Refresh] button to reload the data.

© Welkin Systems Limited


Race Data (1) - 18/01/2004 #3: Trackwork
18/01/2004 # 3 03-339 A S1800 M FG Rtg(Cls) 60-40 (4) 14:00 RIGHT WAY HANDICA
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
Horse Day 17/1 - 11/1 10/1 - 4/1 3/1 - 28/12 27/12 - 21/12 20/12 - 14/1
Shabu Sha Sa Outer Eq. Jog 29.1 23.7; Swi 30.8 29.2 25. Outer Eq. Jog ( Outer Eq. Jo
B290 Fr Swim Aroe Jog Swim; Aroe Jog Swim; Aroe Jog Aroe Jog
Th 27.3 24.2; Aro Swim; Outer Eq Swim; Outer Eq Swim; Outer Eq Swim; Outer
We Aroe Jog Swim; Aroe Jog Swim; Aroe Jog Swim
Tu Outer Eq. Jog ( Swim; Outer Eq Outer Eq. Jog ( Outer Eq. Jog ( Outer Eq. Jo
Mo Aroe Jog Swim; Aroe Jog Aroe Jog Aroe Jog
Su Swim; Aroe Jog Swim
Dongguan Sa Inner Eq. Jog ( Aroe Jog Inner Eq. Jog S1600,10-1
B382 Fr Swim; Inner Eq Inner Eq. Jog 29.0 24.6 Aroe Jog Swim
Th 32.4 26.8 24. Inner Eq. Jog Inner Eq. Jog ( Aroe Jog 34.8 31.7
We Swim; Inner Eq 33.1 30.9 Inner Eq. Jog Aroe Jog Inner Eq. Jo
Tu 31.0 26.4 23. Inner Eq. Jog Inner Eq. Jog Aroe Jog 32.7 31.2
Mo Inner Eq. Jog ( Inner Eq. Jog ( Inner Eq. Jog Aroe Jog Inner Eq. Jo
Su
Cutoff date
17/01/2004
Horizontal
Descending
Vertical
Descending
Refresh

## 5.3.3 Allocation Page (Race Data Form)

The page shows how trainers allocate horses and jockeys in a meeting. You may choose between show allocation of either Horse or Jockey using the option buttons at the top of the page.

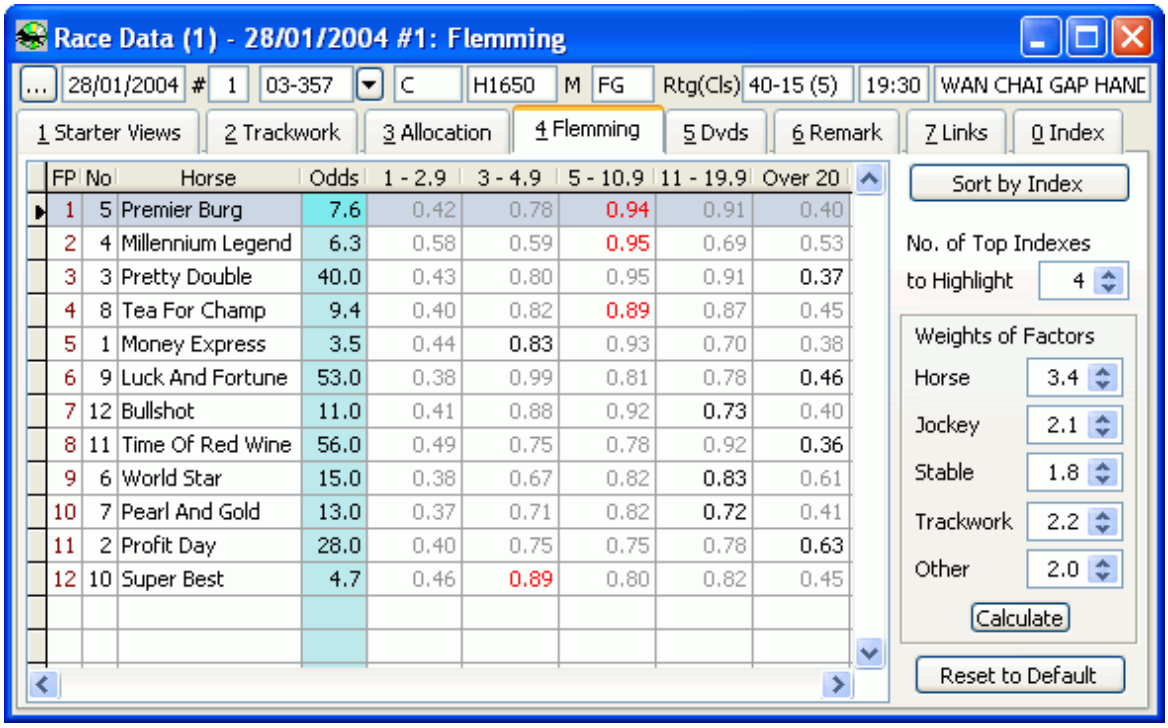

Race Data (1) - 18/01/2004 #3: Allocation
18/01/2004 # 3 03-339 A S1800 M FG Rtg(Cls) 60-40 (4) 14:00 RIGHT WAY HANDICA
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
Horse
Jockey
Stable Race 1 Race 2 Race 3 Race 4 Race 5 Race 6 Race 7 Race 8 Race 9
********** Lucky Lad
A S Cruz Zero In Millennium China Cat Mystic Spir Nilometer Country M Hidden Dra Sure Fit Country Bo
A T Millard Wah Yan M Envoy of L Winning At Little Eleph Champion Splendid P. Noble Man
B K Ng Smiling For Watchman Speed Of L Major Bee
C Fownes Lighting Th Entertainir Sight Chan Thunder M
C H Yip Dynamic P Barracuda Yacht Mast Chinook
C S Shum For Your E Shabu Sha By Memory Miracles
D A Hayes Dongguan Tabbris Tom's First Salut Beau Elegant Fa Songs Of F Spinning C
********** Meridian S
********** All Thrills T
D Cruz Good Good Shenanega Telecom Ty My Fortun Reward
D E Ferrari Natural Mc Elegance Cheerful H Supreme M Momentum Perfect
D Hill No Compro Desert Sto First Knigh Cendamv Peak Powe

© Welkin Systems Limited

## 5.3.4 Flemming Page (Race Data Form)

Flemming Betting Index is a computer rating system based on the Flemming Statistics model. Comprehensive factors, which affect the racing result, are taken into account. You may adjust the weight of such factor based you your own judgment.

Flemming's betting theory regards odds of a decisive figure. Different odds of a horse yield different indexes. In the example race shown below, the Flemming Indexes of Premier Burg were 0.94 and 0.91 in the odd ranges 5-10.9 and 11-19.9 respectively. The actual final odds figure of the horse was 7.6, which fell into the range 5-10.9. Therefore the corresponding index for the horse was 0.94. Other index figures are in grey color showing that they are irrelevant.




| FP | No | Horse | Odds | 1 - 2.9 | 3 - 4.9 | 5 - 10.9 | 11 - 19.9 | Over 20 |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | Premier Burg | 7.6 | 0.42 | 0.78 | 0.94 | 0.91 | 0.40 |
| 2 | 4 | Millennium Legend | 6.3 | 0.58 | 0.59 | 0.95 | 0.69 | 0.53 |
| 3 | 3 | Pretty Double | 40.0 | 0.43 | 0.80 | 0.95 | 0.91 | 0.37 |
| 4 | 8 | Tea For Champ | 9.4 | 0.40 | 0.82 | 0.89 | 0.87 | 0.45 |
| 5 | 1 | Money Express | 3.5 | 0.44 | 0.83 | 0.93 | 0.70 | 0.38 |
| 6 | 9 | Luck And Fortune | 53.0 | 0.38 | 0.99 | 0.81 | 0.78 | 0.46 |
| 7 | 12 | Bullshot | 11.0 | 0.41 | 0.88 | 0.92 | 0.73 | 0.40 |
| 8 | 11 | Time Of Red Wine | 56.0 | 0.49 | 0.75 | 0.78 | 0.92 | 0.36 |
| 9 | 6 | World Star | 15.0 | 0.38 | 0.67 | 0.82 | 0.83 | 0.61 |
| 10 | 7 | Pearl And Gold | 13.0 | 0.37 | 0.71 | 0.82 | 0.72 | 0.41 |
| 11 | 2 | Profit Day | 28.0 | 0.40 | 0.75 | 0.75 | 0.78 | 0.63 |
| 12 | 10 | Super Best | 4.7 | 0.46 | 0.89 | 0.80 | 0.82 | 0.45 |

### Odds Column

Before the races starts, the odds column of the table is blank. You may enter trial figures into the column for analysis. When you enter odds into this column, appropriate betting indexes will be highlighted accordingly. When you update your RaceMate database after a race, the odds in this column will be replaced by the official final odds.

### Sort by Index

Order horse rows by descending order of the betting indexes.

### No. of Top Indexes to Highlight

Number of highest betting indexes to be highlighted.

### Calculate

Recalculate the indexes after changing weights of the factors.

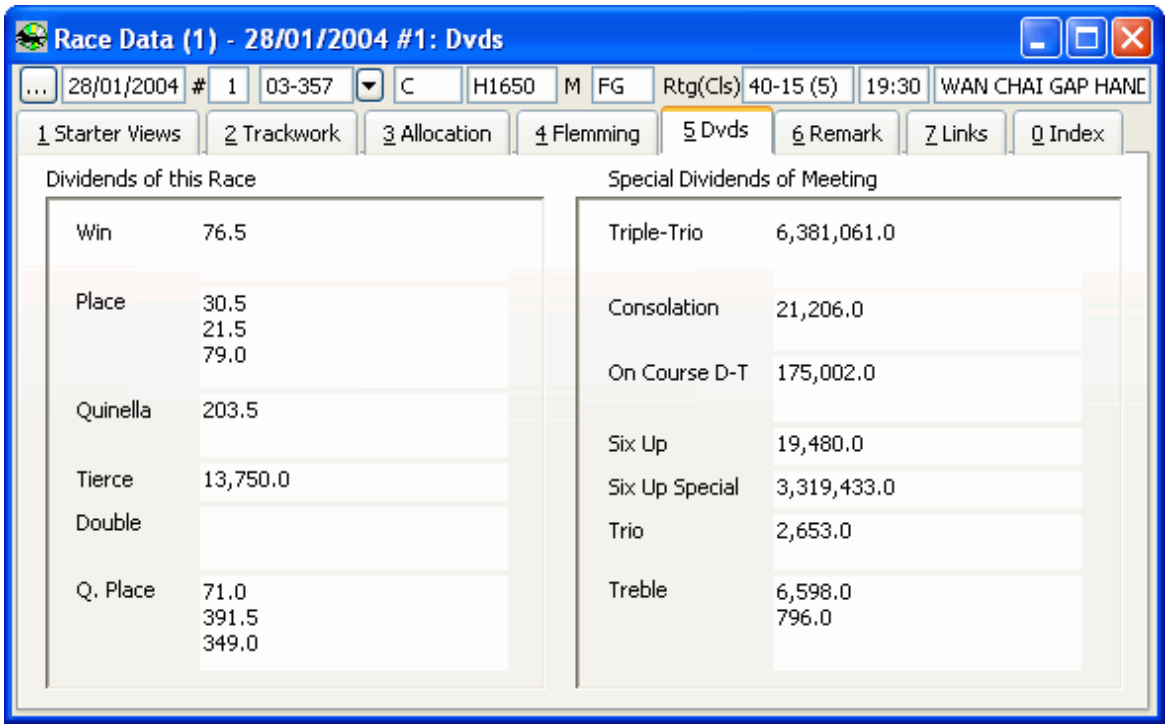
© Welkin Systems Limited

**Reset to Default**

Reset weights of factors to their default values.

### 5.3.5 Dividends Page (Race Data Form)

This page shows dividends information of the race at the left hand side and special dividends of the meeting at the right hand side.




### 5.3.6 Remark Page (Race Data Form)

The Remark page is composed of the following controls:

1. Sectional Time text box

   Sectional time of the leading horse of each section

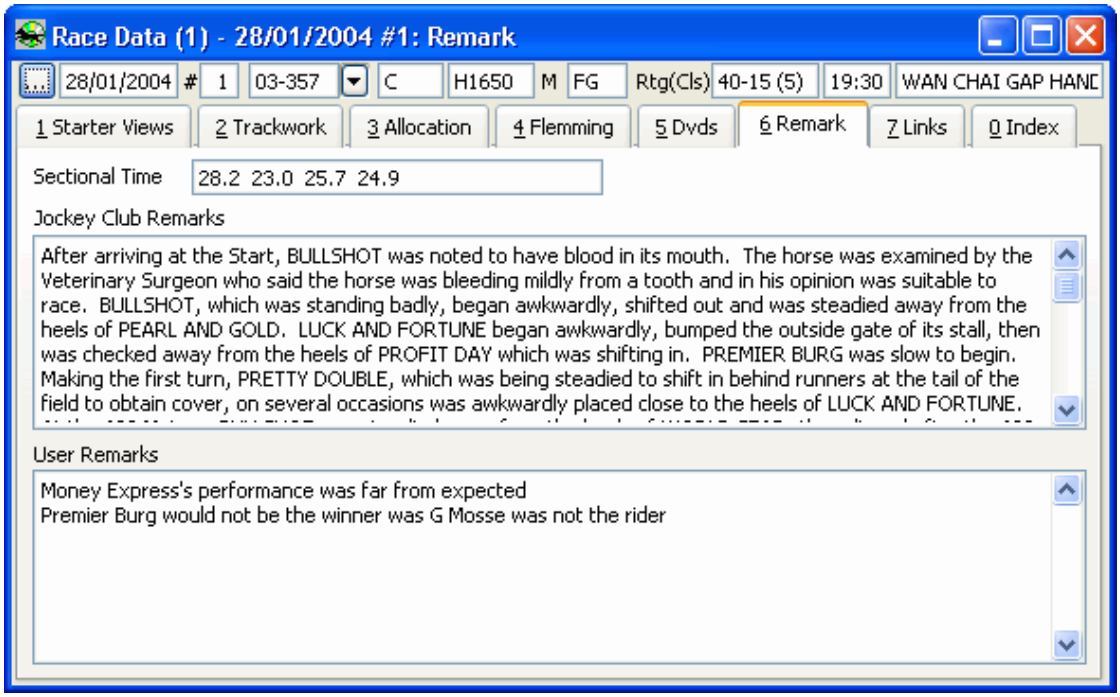
2. Jockey Club Remarks

   Official remarks from the Jockey Club

3. User Remarks

   Your own comments for the race. You must push Ctrl+E (or click on the [Edit] button on the toolbar) to put the form into the Edit Mode before inputting and push Ctrl+S to save after finishing.

© Welkin Systems Limited


Race Data (1) - 28/01/2004 #1: Remark
28/01/2004
#
1
03-357
C
H1650
M
FG
Rtg(Cls)
40-15 (5)
19:30
WAN CHAI GAP HANC
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
Sectional Time
28.2 23.0 25.7 24.9
Jockey Club Remarks
After arriving at the Start, BULLSHOT was noted to have blood in its mouth. The horse was examined by the Veterinary Surgeon who said the horse was bleeding mildly from a tooth and in his opinion was suitable to race. BULLSHOT, which was standing badly, began awkwardly, shifted out and was steadied away from the heels of PEARL AND GOLD. LUCK AND FORTUNE began awkwardly, bumped the outside gate of its stall, then was checked away from the heels of PROFIT DAY which was shifting in. PREMIER BURG was slow to begin. Making the first turn, PRETTY DOUBLE, which was being steadied to shift in behind runners at the tail of the field to obtain cover, on several occasions was awkwardly placed close to the heels of LUCK AND FORTUNE.
User Remarks
Money Express's performance was far from expected
Premier Burg would not be the winner was G Mosse was not the rider

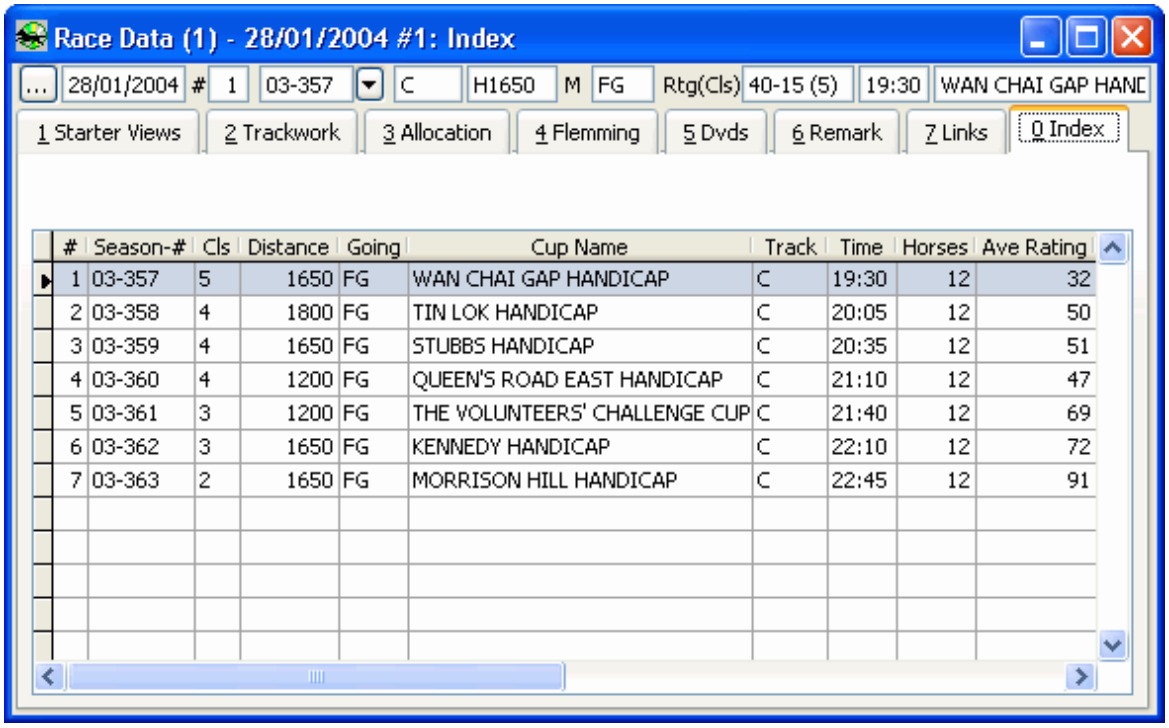
### 5.3.7 Links Page (Race Data Form)

Please see MultimediaResources, Autolinks and Managing Your Own Multimedia Resources

### 5.3.8 Index Page (Race Data Form)

This page lists all races of a meeting. You may highlight a race on the list and push the [Enter] key or double-click to jump back to the page which is active before opening the Index page. Information about the race highlighted in the Index Page will be displayed. If there was no page active before the Index page was opened, the Starter Views Page will be opened.

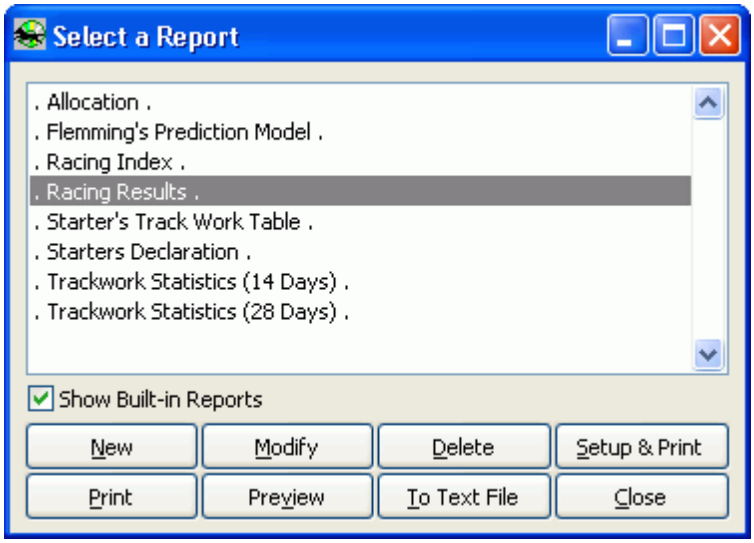
© Welkin Systems Limited


Race Data (1) - 28/01/2004 #1: Index
28/01/2004 # 1 03-357 C H1650 M FG Rtg(Cls) 40-15 (5) 19:30 WAN CHAI GAP HANC
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index

| # | Season-# | Cls | Distance | Going | Cup Name | Track | Time | Horses | Ave Rating |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 03-357 | 5 | 1650 | FG | WAN CHAI GAP HANDICAP | C | 19:30 | 12 | 32 |
| 2 | 03-358 | 4 | 1800 | FG | TIN LOK HANDICAP | C | 20:05 | 12 | 50 |
| 3 | 03-359 | 4 | 1650 | FG | STUBBS HANDICAP | C | 20:35 | 12 | 51 |
| 4 | 03-360 | 4 | 1200 | FG | QUEEN'S ROAD EAST HANDICAP | C | 21:10 | 12 | 47 |
| 5 | 03-361 | 3 | 1200 | FG | THE VOLUNTEERS' CHALLENGE CUP | C | 21:40 | 12 | 69 |
| 6 | 03-362 | 3 | 1650 | FG | KENNEDY HANDICAP | C | 22:10 | 12 | 72 |
| 7 | 03-363 | 2 | 1650 | FG | MORRISON HILL HANDICAP | C | 22:45 | 12 | 91 |

## 5.3.9 Print Race Data (Race Data Form)

Click on the [Print] button of the toolbar to displays the following dialog box. The box shows a list of pre-defined reports. You can select a report and click one of the buttons to perform the corresponding action. You may design your own report for a Starter View. See Creating a Report for a View for details.




**New**

Opens the Report Designer and allows you to create a new report.

© Welkin Systems Limited

**Modify**

Opens the Report Designer and allows you to edit the selected report. Delete Deletes the selected report.

**Setup & Print**

Displays the Print Setup dialog box in which you can select printer, paper size, and orientation of printing.

**Print**

Outputs the report to your printer.

**Preview**

Displays the preview of the selected report.

**To Text File**

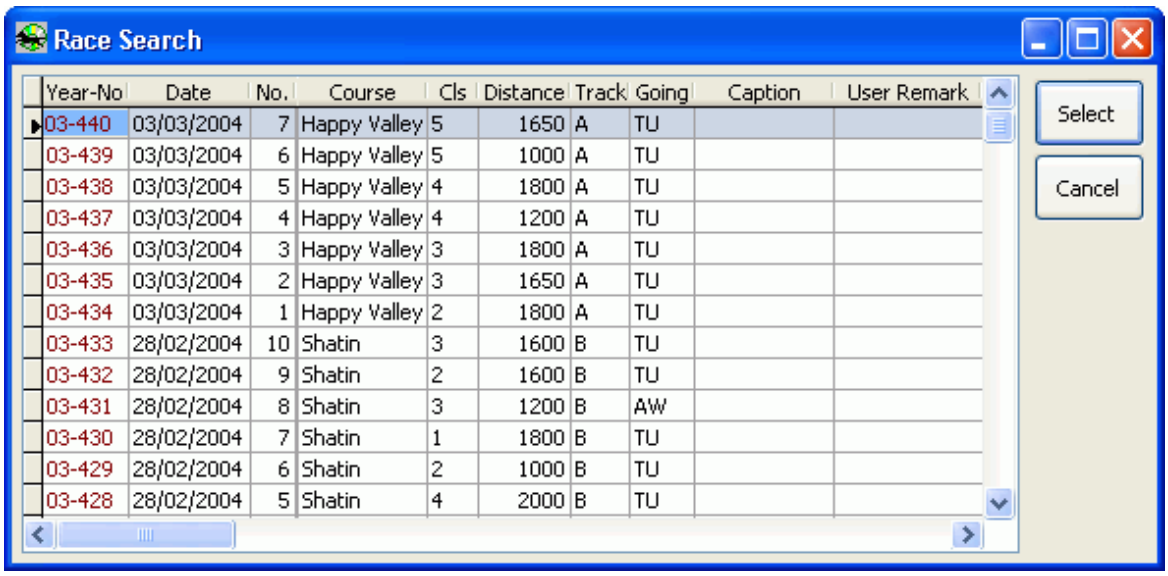
Clicking this button will print to a text file. You will be asked to input the filename.

**Close**

Closes this dialog box.

## 5.4 Race Search Form

The Race Search Form helps you quickly locate a race. You can do an incremental search on any column of the Race Search Form or carry out a multicolumn sort. After you have located a race, press Enter or double-click on the race to open it in the Race Data Form.

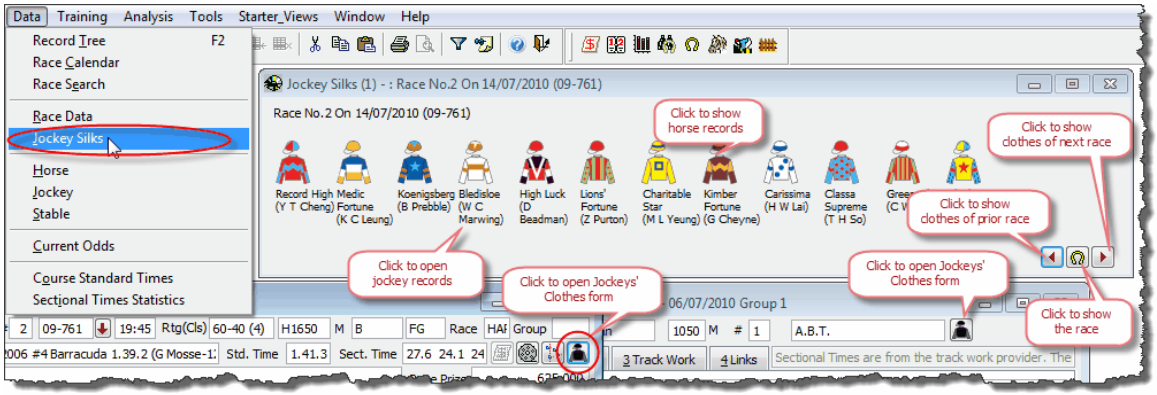



© Welkin Systems Limited

## 5.5 Jockey Silks

The Jockeys Silks form shows the clothes of horse riders. The clothes are unique to individual horses. You can open this form in one of the following ways.

1. Choose [Jockey Silks] from the [Data] menu
2. Click on the [Jockey Silks] button on the Race Data form
3. Click on the [Jockey Silks] button on the Barrier Trial form




On the form you may:

1. Click on Jockey Silk picture to open the Horse form
2. Click on jockey names or horse names to open the Jockey form
3. Click on the [Next Race] and [Prior Race] button to navigate around races
4. Click on the [Show Race Form] button to open the Race Data form

Please note that Jockey Silks for newly arrived horses or horses retired before season 2009-2010 may be missing.

## 5.6 Horse Form

The Horse form is usually opened by:

- double-clicking on a horse's name in any form
- clicking Data - Horse on the menu bar

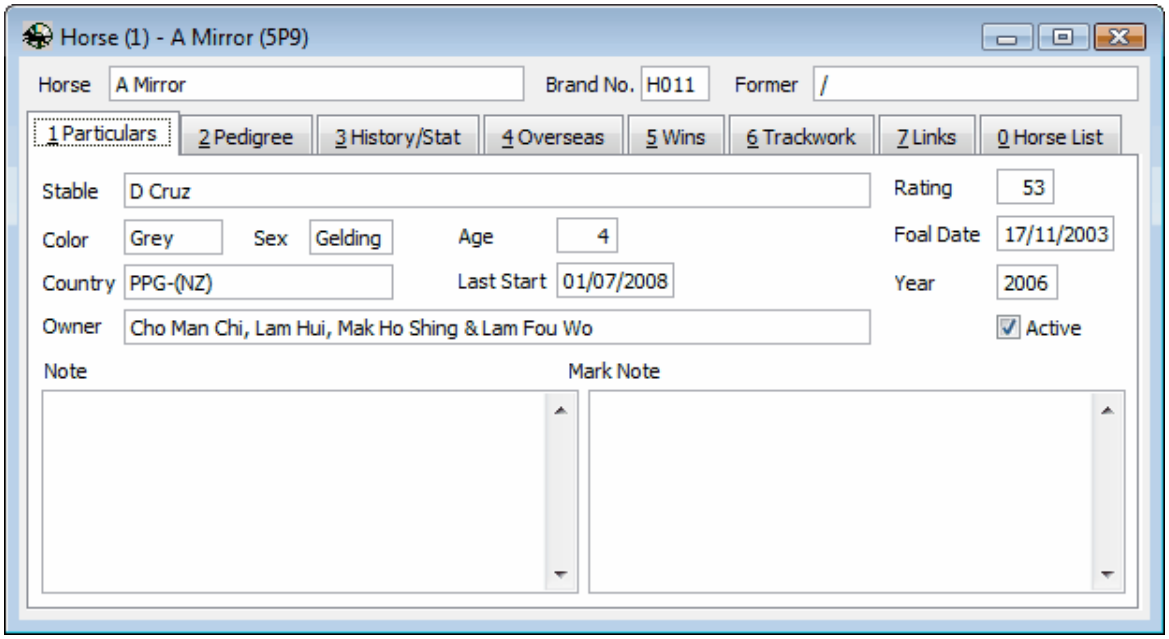
**>>> Subtopics of this section <<<**

- Particulars Page (Horse Form)
- Pedigree Page (Horse Form)
- Past Record and Comprehensive Statistics for (Horse Form)
- Overseas Page (Horse Form)
- Wins Page (Horse Form)
- Trackwork (Horse Form)
- Links Page (Horse Form)
- Horse List Page (Horse Form)

© Welkin Systems Limited

## 5.6.1 Particulars Page (Horse Form)

This page show general information about a horse including trainer, rating, origin, and so on.

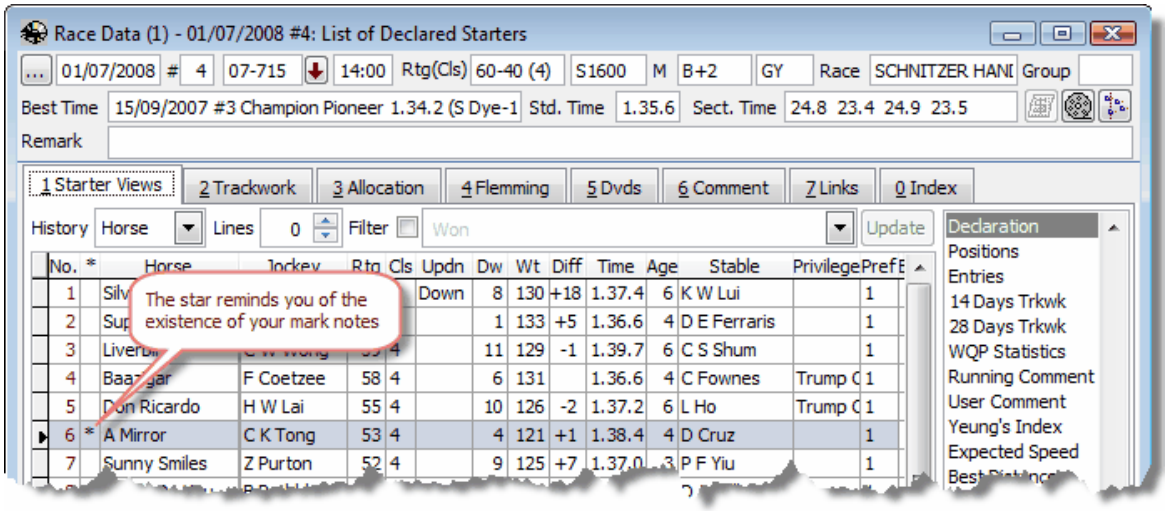



### Inputting Your Notes

You may input your own comments and notes about the horse into the edit box at the left bottom of the page. Before editing, you may need to press Ctrl+E (or push the Edit button of the toolbar) to put the form into the Edit Mode. After finishing editing, press Ctrl+S to save your change.

### Editing Mark Notes

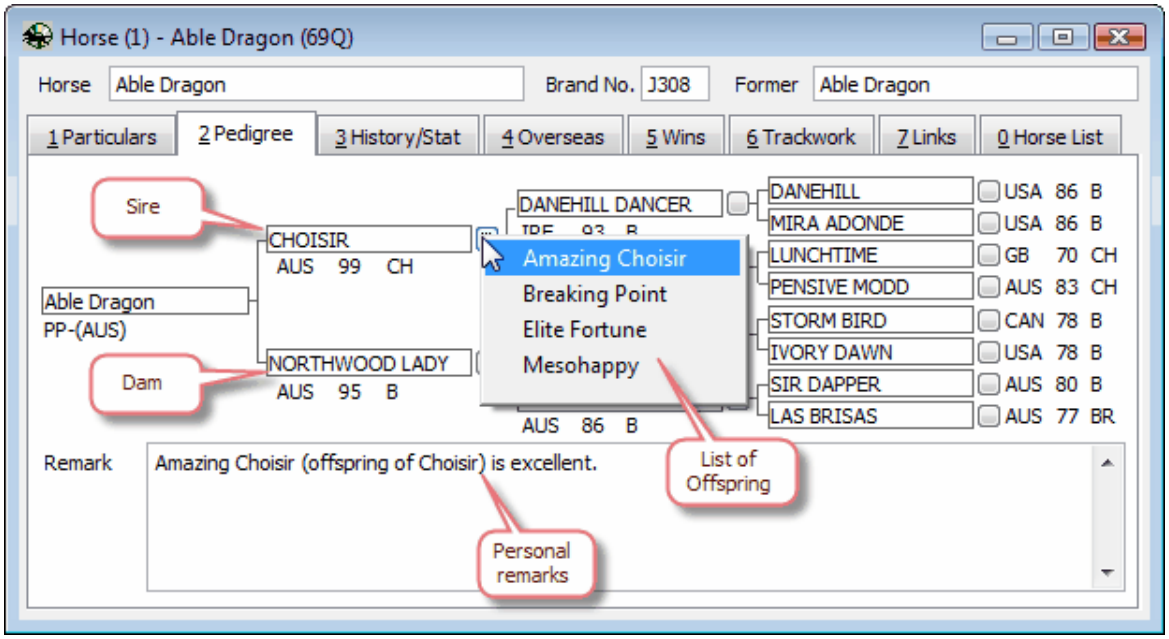
Input your Mark Note into the edit box at the right bottom of the page. Both Mark Note and Note are your comments about the horse. However, when you have entered Mark Note for a horse, there will be a star shown in the "*" column of the Declaration View to remind you of the existence of the Mark Note.


Race Data (1) - 01/07/2008 #4: List of Declared Starters
01/07/2008 # 4 07-715 14:00 Rtg(Cls) 60-40 (4) S1600 M B+2 GY Race SCHNITZER HANI Group
Best Time 15/09/2007 #3 Champion Pioneer 1.34.2 (S Dye-1 Std. Time 1.35.6 Sect. Time 24.8 23.4 24.9 23.5
Remark
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Comment
7 Links
0 Index
History Horse Lines 0 Filter Won Update
No. * Horse Jockey Rtg Cls Updn Dw Wt Diff Time Age Stable Privilege Pref
1 Down 8 130 +18 1.37.4 6 K W Lui 1
2 1 133 +5 1.36.6 4 D E Ferraris 1
3 11 129 -1 1.39.7 6 C S Shum 1
4 F Coetzee 58 4 6 131 1.36.6 4 C Fownes Trump C 1
5 H W Lai 55 4 10 126 -2 1.37.2 6 L Ho Trump C 1
6 * A Mirror C K Tong 53 4 4 121 +1 1.38.4 4 D Cruz 1
7 Sunny Smiles Z Purton 52 4 9 125 +7 1.37.0 3 P F Yiu 1
The star reminds you of the existence of your mark notes
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
User Comment
Yeung's Index
Expected Speed

© Welkin Systems Limited

### 5.6.2 Pedigree Page (Horse Form)

Pedigree tree of 4 generations is show on the Pedigree page. You can click on the small button at the right of the horse name to show a list of horses which are the offspring of the horse.




### 5.6.3 Past Record and Comprehensive Statistics for (Horse Form)

The purposes of this page are:

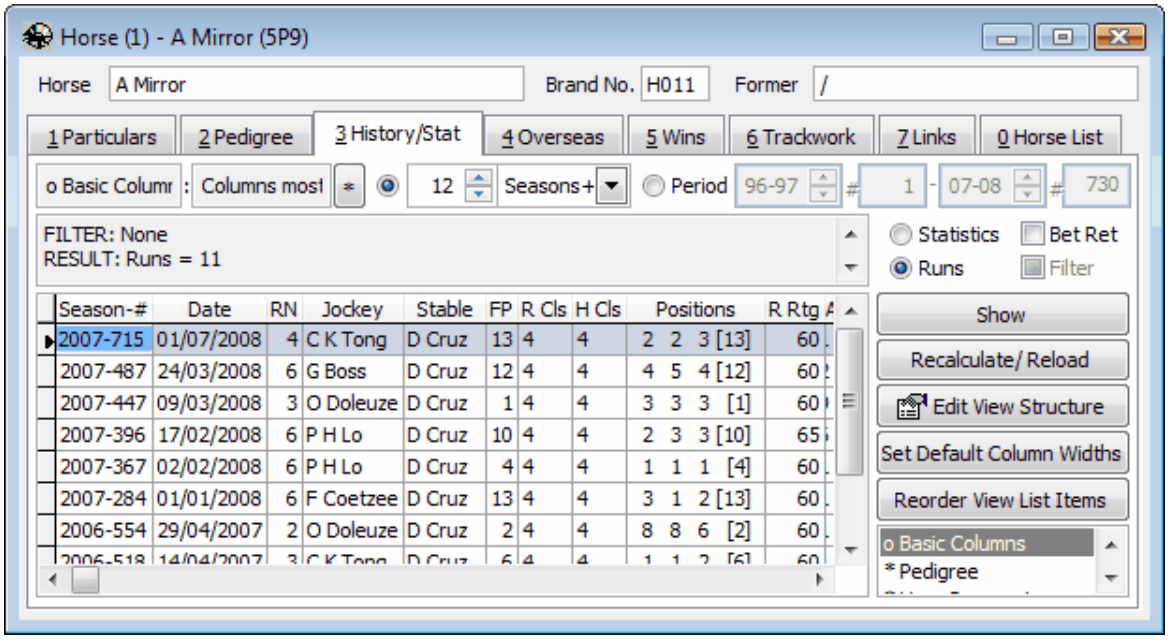
1. Display complete or filtered past records
2. Define comprehensive criteria to carry statistics and queries on past records
3. Carry out comprehensive statistics

Further more you may do the following on the form:

- Create your own view with the data columns that you need.
- Add or remove columns from existing views.
- Setting a period to limit the records retrieved.
- Create a filter to retrieve only the records which meet the criteria you want.

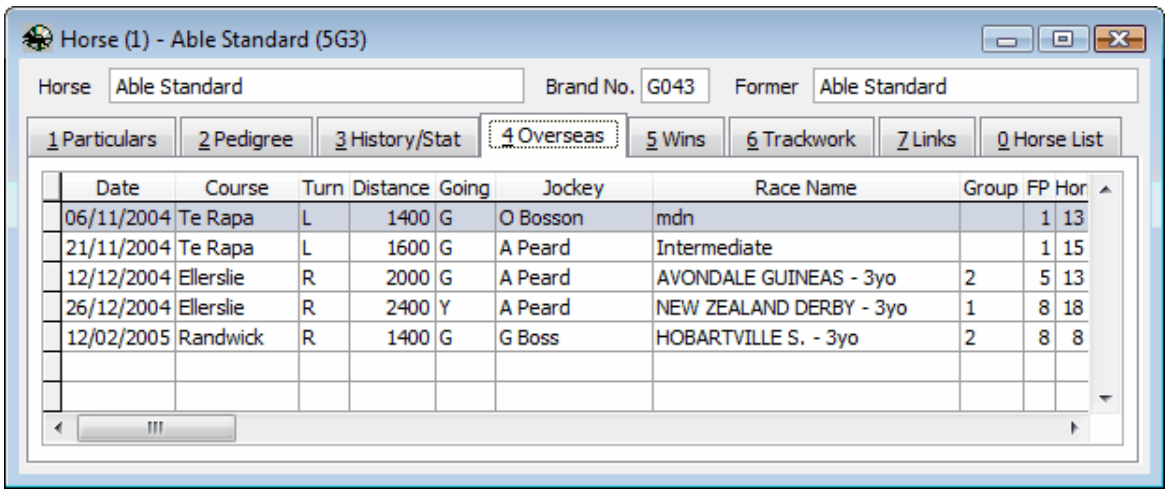
Detailed operation instructions and functions of each elements of the page are shown on the page "Comprehensive Statistics and Record Query" (under the chapter Statistics and Data Analysis).

© Welkin Systems Limited




## 5.6.4 Overseas Page (Horse Form)

This page displays races of the horse before coming to Hong Kong.

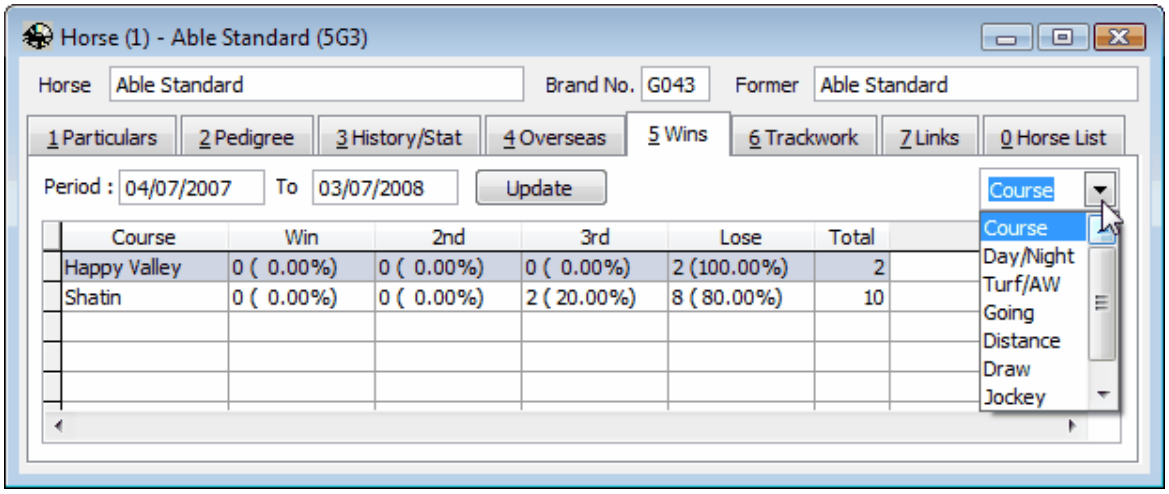

Horse (1) - Able Standard (5G3)
Horse Able Standard
Brand No. G043
Former Able Standard
1 Particulars
2 Pedigree
3 History/Stat
4 Overseas
5 Wins
6 Trackwork
7 Links
0 Horse List
Date Course Turn Distance Going Jockey Race Name Group FP Hor
06/11/2004 Te Rapa L 1400 G O Bosson mdn 1 13
21/11/2004 Te Rapa L 1600 G A Peard Intermediate 1 15
12/12/2004 Ellerslie R 2000 G A Peard AVONDALE GUINEAS - 3yo 2 5 13
26/12/2004 Ellerslie R 2400 Y A Peard NEW ZEALAND DERBY - 3yo 1 8 18
12/02/2005 Randwick R 1400 G G Boss HOBARTVILLE S. - 3yo 2 8 8

## 5.6.5 Wins Page (Horse Form)

This page shows breakdowns of number of wins under different conditions. The following picture show a breakdown by course. You may choose any of the following breakdown settings:

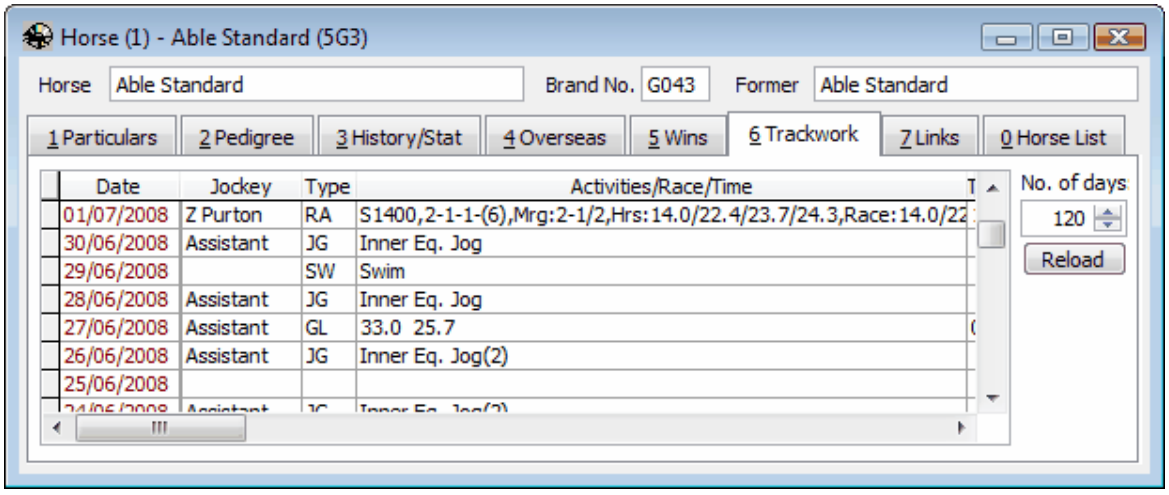
- Course
- Day/Night
- Turf/AW
- Going
- Distance
- Draw
- Jockey
- Track

© Welkin Systems Limited




## 5.6.6 Trackwork (Horse Form)

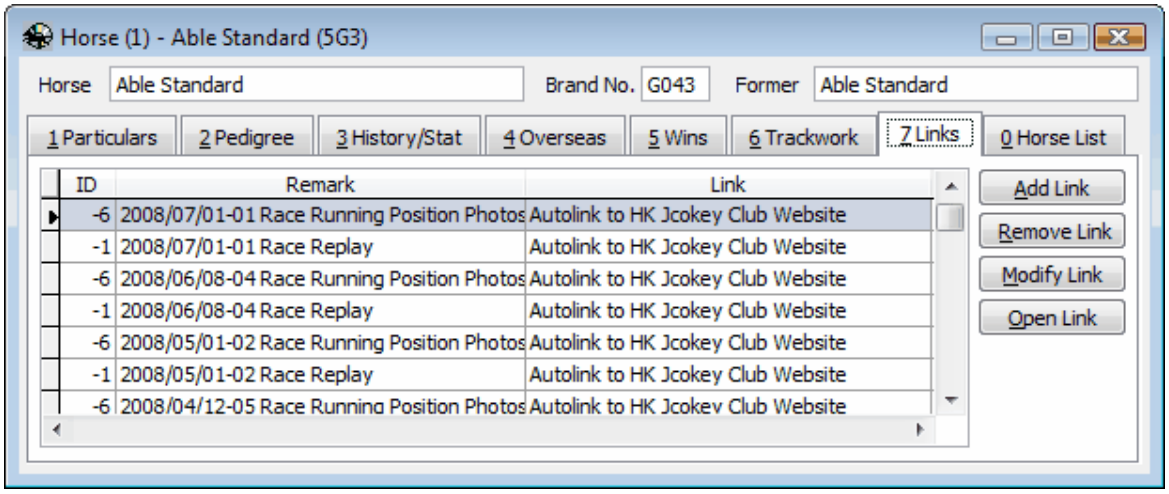
This page shows track work (and races) of the horse. You may set the number of days to be retrieved into the page.




## 5.6.7 Links Page (Horse Form)

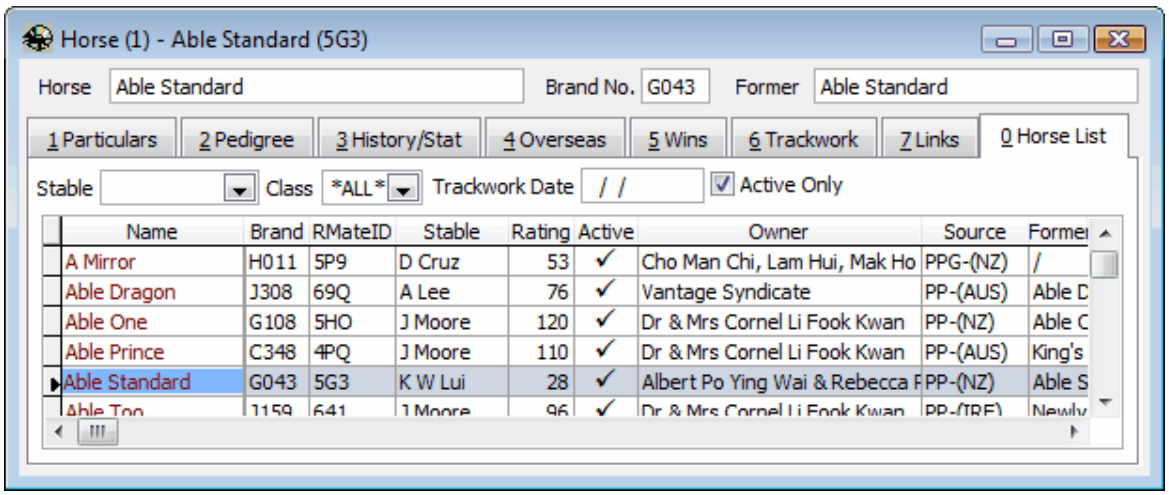
You may open videos of the Jockey Club by double-clicking on the links in the table. The availability of the videos depends on their existence on the server of the Jockey Club. For management about the links, please see the Multimedia Resource section for details.

© Welkin Systems Limited


Horse (1) - Able Standard (5G3)
Horse Able Standard
Brand No. G043
Former Able Standard
1 Particulars
2 Pedigree
3 History/Stat
4 Overseas
5 Wins
6 Trackwork
7 Links
0 Horse List
ID
Remark
Link
-6 2008/07/01-01 Race Running Position Photos Autolink to HK Jcokey Club Website
-1 2008/07/01-01 Race Replay Autolink to HK Jcokey Club Website
-6 2008/06/08-04 Race Running Position Photos Autolink to HK Jcokey Club Website
-1 2008/06/08-04 Race Replay Autolink to HK Jcokey Club Website
-6 2008/05/01-02 Race Running Position Photos Autolink to HK Jcokey Club Website
-1 2008/05/01-02 Race Replay Autolink to HK Jcokey Club Website
-6 2008/04/12-05 Race Running Position Photos Autolink to HK Jcokey Club Website
Add Link
Remove Link
Modify Link
Open Link

### 5.6.8 Horse List Page (Horse Form)

This page shows all horses (including inactive) with other useful information. Please see the Incremental Search about how to locate a horse on the page.


Horse (1) - Able Standard (5G3)
Horse Able Standard
Brand No. G043
Former Able Standard
1 Particulars
2 Pedigree
3 History/Stat
4 Overseas
5 Wins
6 Trackwork
7 Links
0 Horse List
Stable
Class *ALL*
Trackwork Date / /
Active Only
Name
Brand
RMateID
Stable
Rating
Active
Owner
Source
Former
A Mirror H011 5P9 D Cruz 53 ✓ Cho Man Chi, Lam Hui, Mak Ho PPG-(NZ) /
Able Dragon J308 69Q A Lee 76 ✓ Vantage Syndicate PP-(AUS) Able D
Able One G108 5HO J Moore 120 ✓ Dr & Mrs Cornel Li Fook Kwan PP-(NZ) Able C
Able Prince C348 4PQ J Moore 110 ✓ Dr & Mrs Cornel Li Fook Kwan PP-(AUS) King's
Able Standard G043 5G3 K W Lui 28 ✓ Albert Po Ying Wai & Rebecca F PP-(NZ) Able S

## 5.7 Jockey Form

The Jockey form is usually opened by:

- double-clicking on a jockey's name in any form
- clicking Data - Jockey on the menu bar

**>>> Subtopics of this section <<<**

- Past Record and Comprehensive Statistics (Jockey Form)
- Wins Page (Jockey Form)
- Particulars/Notes Page (Jockey Form)
- Links Page (Jockey Form)
- Jockeys List Page (Jockey Form)

### 5.7.1 Past Record and Comprehensive Statistics (Jockey Form)

The purposes of this page are:

1. Display complete or filtered past records

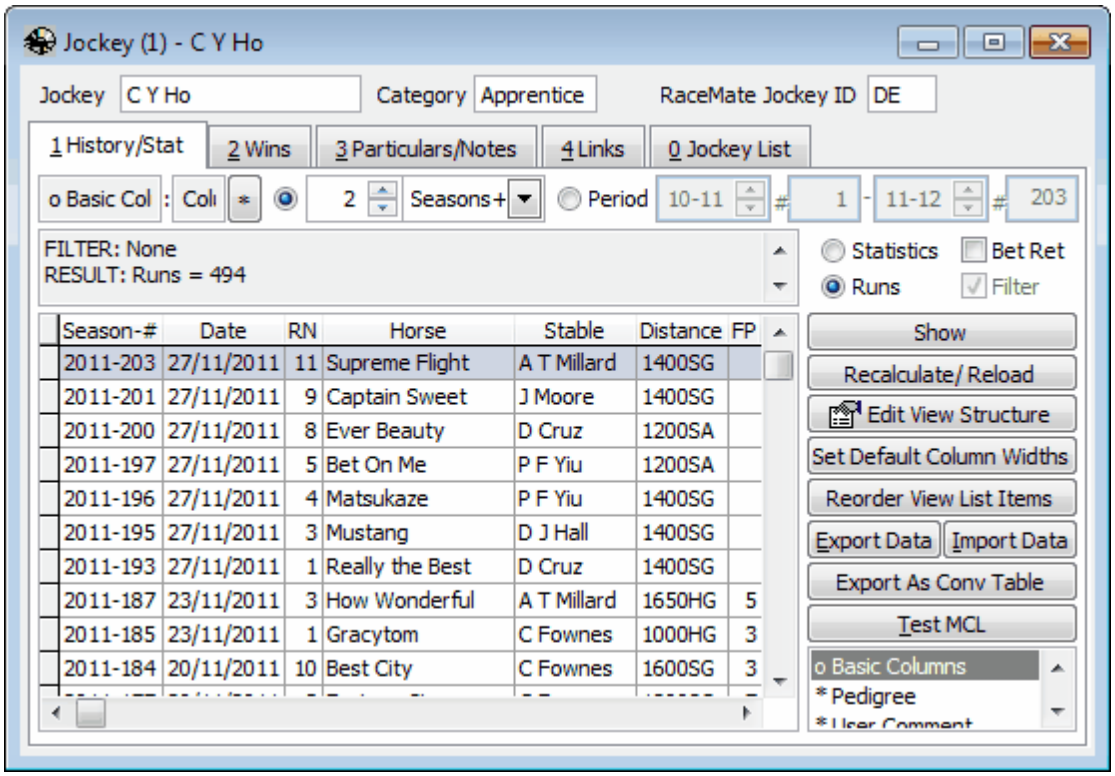
© Welkin Systems Limited

2. Define comprehensive criteria to carry statistics and queries on past records
3. Carry out comprehensive statistics

Further more you may do the following on the form:

- Create your own view with the data columns that you need.
- Add or remove columns from existing views.
- Setting a period to limit the records retrieved.
- Create a filter to retrieve only the records which meet the criteria you want.

Detailed operation instructions and functions of each elements of the page are shown on the page "Comprehensive Statistics and Record Query" (under the chapter Statistics and Data Analysis).

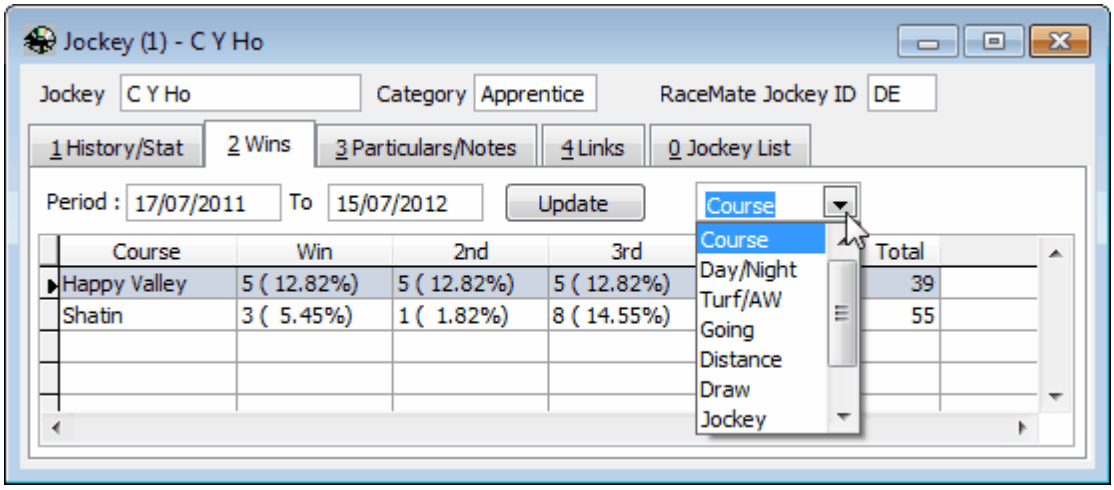



## 5.7.2 Wins Page (Jockey Form)

This page shows breakdowns of number of wins under different conditions. The following picture show a breakdown by course. You may choose any of the following breakdown settings:

- Course
- Day/Night
- Turf/AW
- Going
- Distance
- Draw
- Jockey
- Track

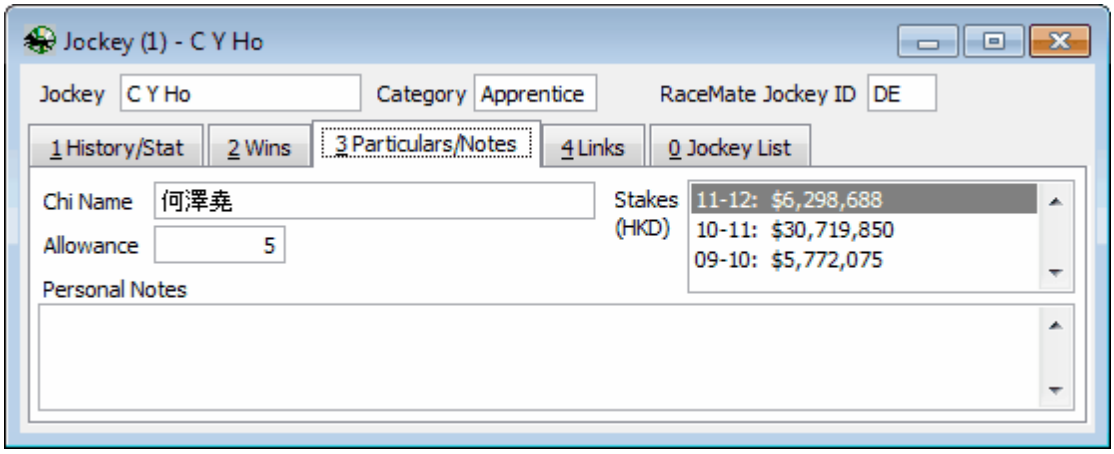
© Welkin Systems Limited




## 5.7.3 Particulars/Notes Page (Jockey Form)

You can input your personal note for the jockey on this page. You need to push [Ctrl+E] (or the Edit button on the toolbar) to change to the Edit mode before you can input your notes. After edit finished, push [Ctrl+S] to save your change.

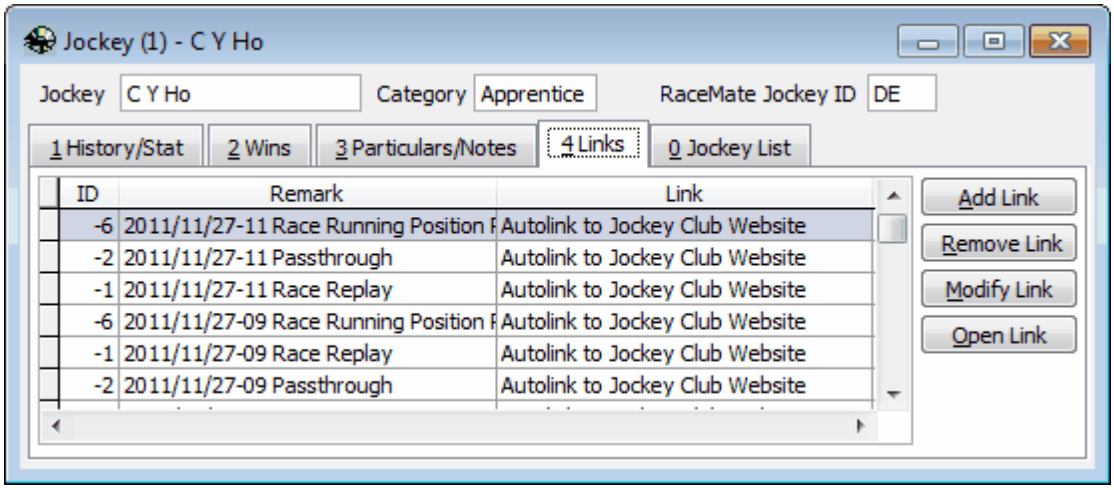
The page also shows the Chinese name, allowance (no such item in Macau edition) and stakes got past seasons.




## 5.7.4 Links Page (Jockey Form)

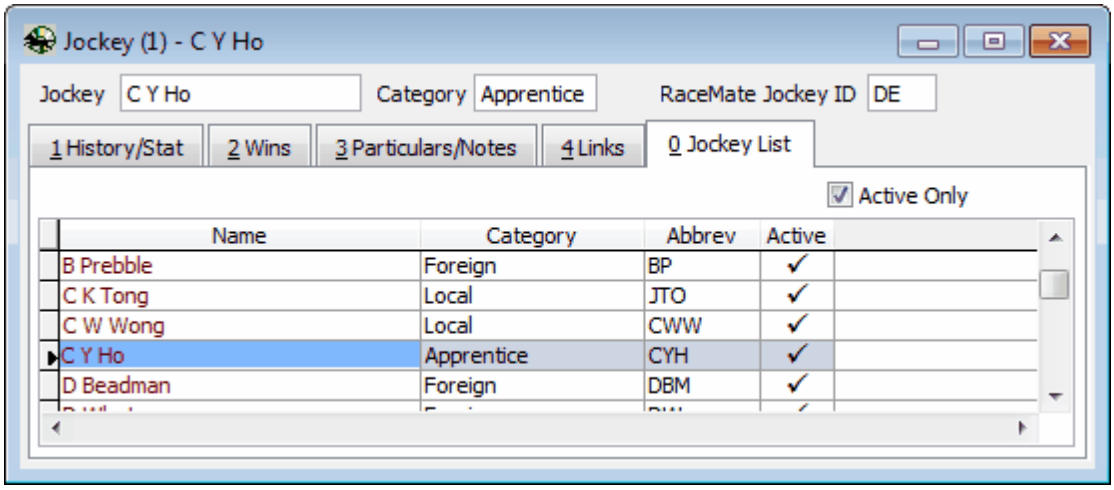
You may open videos of the Jockey Club by double-clicking on the links in the table. The availability of the videos depends on their existence on the server of the Jockey Club. For management about the links, please see the Multimedia Resource section for details.

© Welkin Systems Limited


Jockey (1) - C Y Ho
Jockey C Y Ho
Category Apprentice
RaceMate Jockey ID DE
1 History/Stat
2 Wins
3 Particulars/Notes
4 Links
0 Jockey List
ID
Remark
Link
-6 2011/11/27-11 Race Running Position F Autolink to Jockey Club Website
-2 2011/11/27-11 Passthrough Autolink to Jockey Club Website
-1 2011/11/27-11 Race Replay Autolink to Jockey Club Website
-6 2011/11/27-09 Race Running Position F Autolink to Jockey Club Website
-1 2011/11/27-09 Race Replay Autolink to Jockey Club Website
-2 2011/11/27-09 Passthrough Autolink to Jockey Club Website
Add Link
Remove Link
Modify Link
Open Link

### 5.7.5 Jockeys List Page (Jockey Form)

This page shows all horses (including inactive) with other useful information. Please see the Incremental Search about how to locate a jockey on the page.




## 5.8 Stable Form

The Stable form is usually opened by:

- double-clicking on a stable's name in any form
- clicking Data - Stable on the menu bar

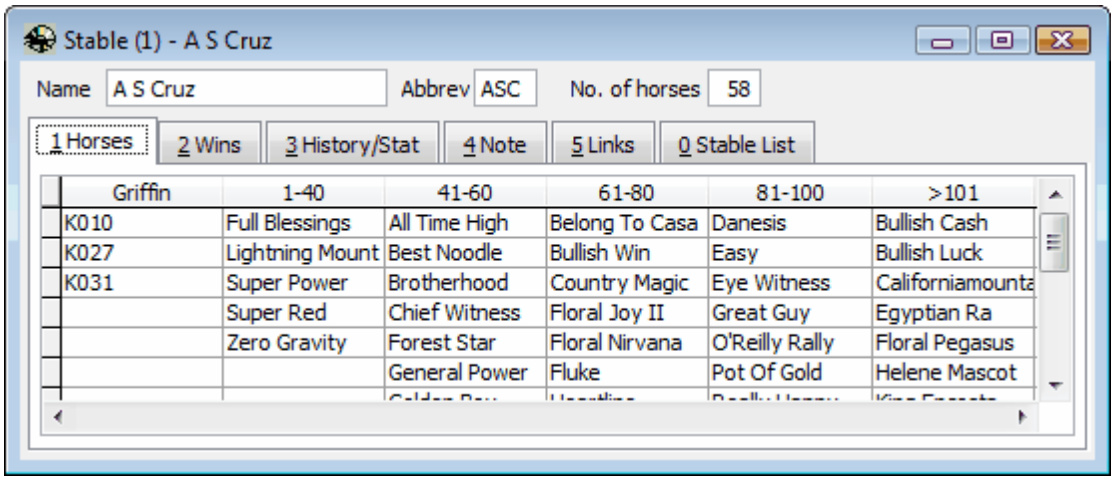
**>>> Subtopics of this section <<<**

- Horses in Stable (Stable Form)
- Wins Page (Stable Form)
- Past Record and Comprehensive Statistics (Stable Form)
- Notes Page (Stable Form)
- Links Page (Stable Form)
- Stables List Page (Stable Form)

© Welkin Systems Limited

### 5.8.1 Horses in Stable (Stable Form)

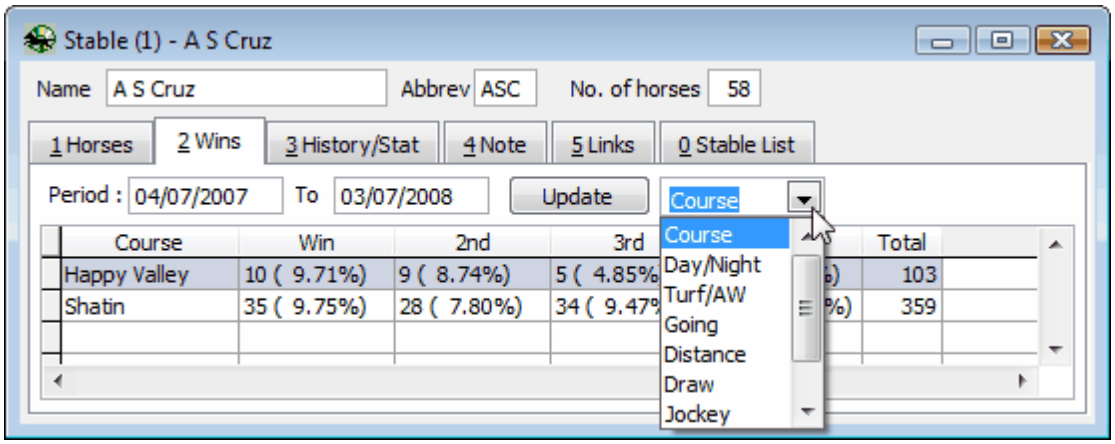
This page show all horses of the stable. They are grouped according to their ratings.


Stable (1) - A S Cruz
Name A S Cruz
Abbrev ASC
No. of horses 58
1 Horses
2 Wins
3 History/Stat
4 Note
5 Links
0 Stable List
Griffin 1-40 41-60 61-80 81-100 >101
K010 Full Blessings All Time High Belong To Casa Danesis Bullish Cash
K027 Lightning Mount Best Noodle Bullish Win Easy Bullish Luck
K031 Super Power Brotherhood Country Magic Eye Witness Californiamounta
Super Red Chief Witness Floral Joy II Great Guy Egyptian Ra
Zero Gravity Forest Star Floral Nirvana O'Reilly Rally Floral Pegasus
General Power Fluke Pot Of Gold Helene Mascot

### 5.8.2 Wins Page (Stable Form)

This page shows breakdowns of number of wins under different conditions. The following picture show a breakdown by course. You may choose any of the following breakdown settings:

- Course
- Day/Night
- Turf/AW
- Going
- Distance
- Draw
- Jockey
- Track




### 5.8.3 Past Record and Comprehensive Statistics (Stable Form)

The purposes of this page are:

1. Display complete or filtered past records
2. Define comprehensive criteria to carry statistics and queries on past records
3. Carry out comprehensive statistics

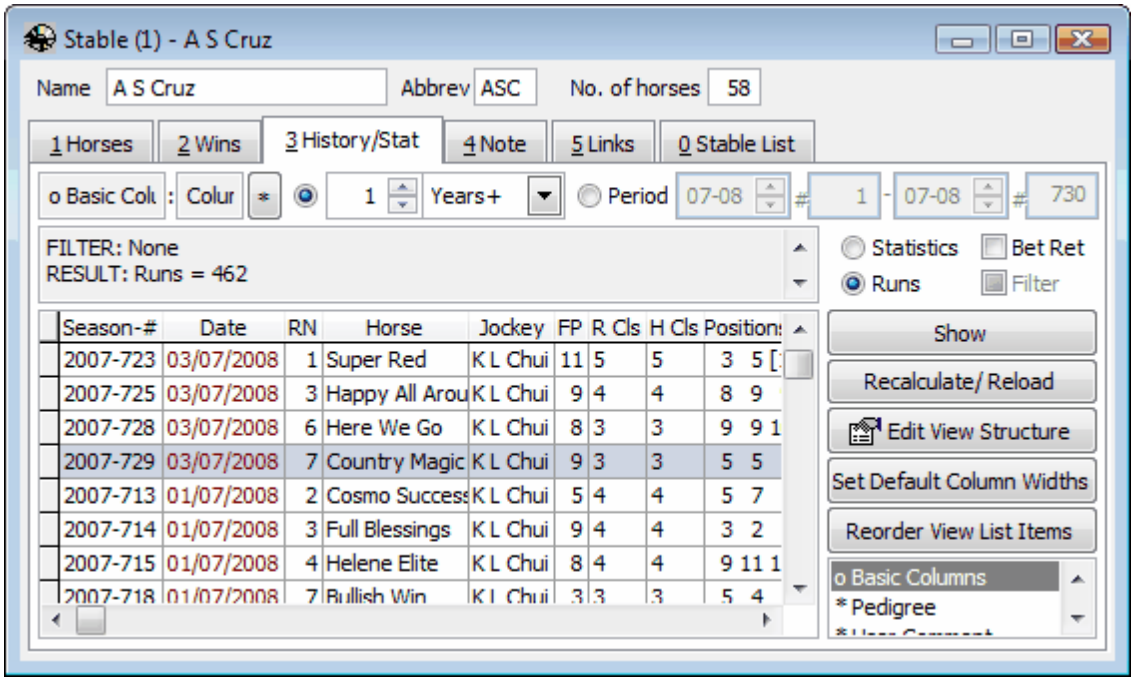
Further more you may do the following on the form:

- Create your own view with the data columns that you need.

© Welkin Systems Limited

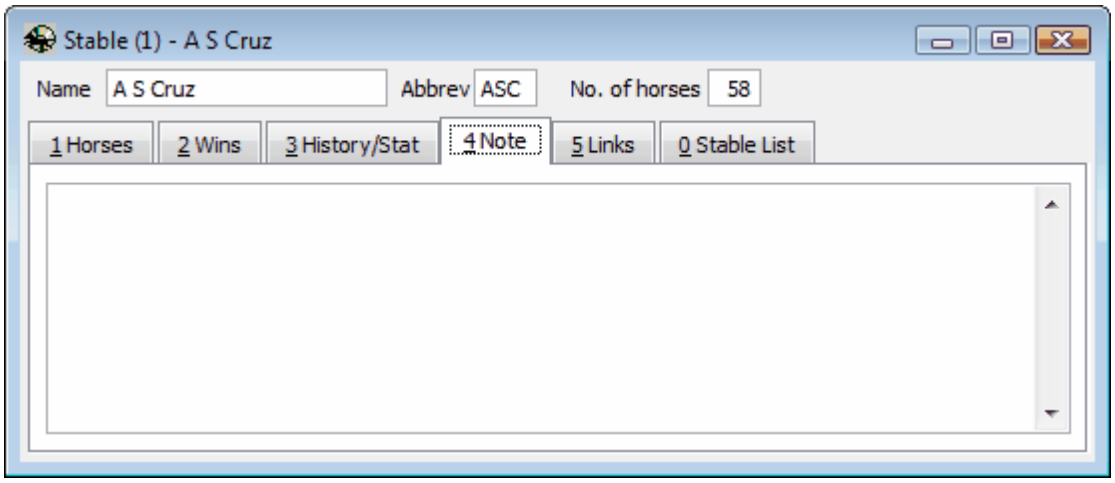
- Add or remove columns from existing views.
- Setting a period to limit the records retrieved.
- Create a filter to retrieve only the records which meet the criteria you want.

Detailed operation instructions and functions of each elements of the page are shown on the page "Comprehensive Statistics and Record Query" (under the chapter Statistics and Data Analysis).




## 5.8.4 Notes Page (Stable Form)

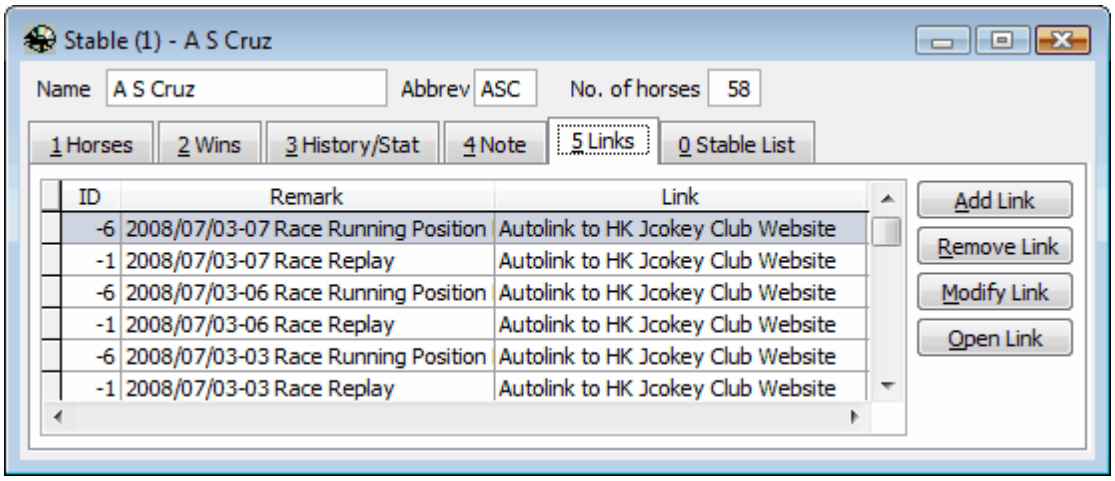
You can input your personal note for the stable on this page. You need to push [Ctrl+E] (or the Edit button on the toolbar) to change to the Edit mode before you can input your notes. After edit finished, push [Ctrl+S] to save your change.




## 5.8.5 Links Page (Stable Form)

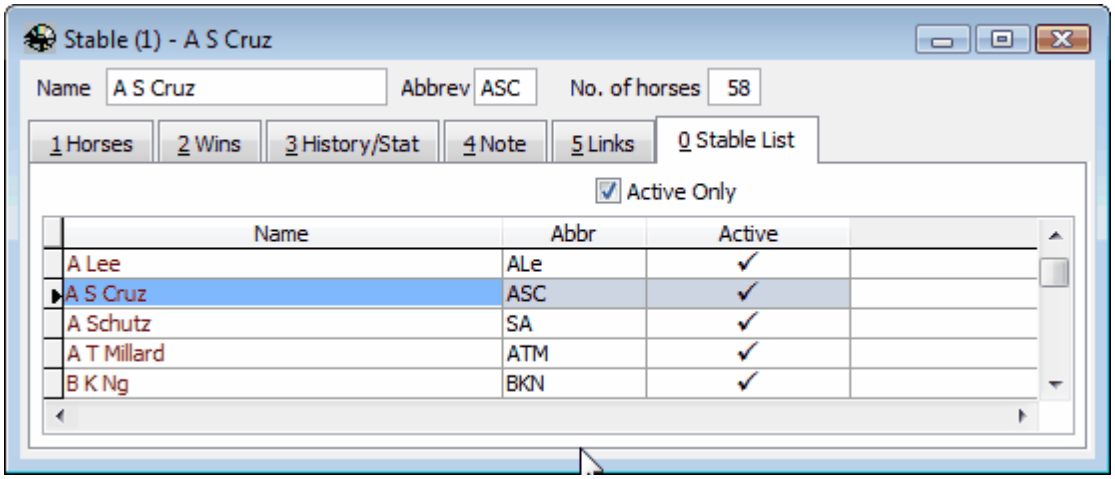
You may open videos of the Jockey Club by double-clicking on the links in the table. The availability of the videos depends on their existence on the server of the Jockey Club. For management about the links, please see the Multimedia Resource section for details.

© Welkin Systems Limited


Stable (1) - A S Cruz
Name A S Cruz
Abbrev ASC
No. of horses 58
1 Horses
2 Wins
3 History/Stat
4 Note
5 Links
0 Stable List
ID
Remark
Link
-6 2008/07/03-07 Race Running Position Autolink to HK Jcokey Club Website
-1 2008/07/03-07 Race Replay Autolink to HK Jcokey Club Website
-6 2008/07/03-06 Race Running Position Autolink to HK Jcokey Club Website
-1 2008/07/03-06 Race Replay Autolink to HK Jcokey Club Website
-6 2008/07/03-03 Race Running Position Autolink to HK Jcokey Club Website
-1 2008/07/03-03 Race Replay Autolink to HK Jcokey Club Website
Add Link
Remove Link
Modify Link
Open Link

### 5.8.6 Stables List Page (Stable Form)

This page shows all horses (including inactive) with other useful information. Please see the Incremental Search about how to locate a stable on the page.




## 5.9 Current Odds from Jockey Club

To get access to the online current odds, you may:

1. Click on the [Online Odds] button on the Race Data form or Race Calendar form
2. Select [Current Odds] from the [Data] menu

RaceMate will open the Jockey Club's online odds web page. Your computer must be connecting to the Internet.

## 5.10 Daily Track Work and Barrier Trial

You may do one of the following to open the Daily Track Work and Barrier Trial form:

- Select [Daily Track Work and Barrier Trial] from the [Training] menu of RaceMate
- Click on a barrier trial group on the Race Calendar

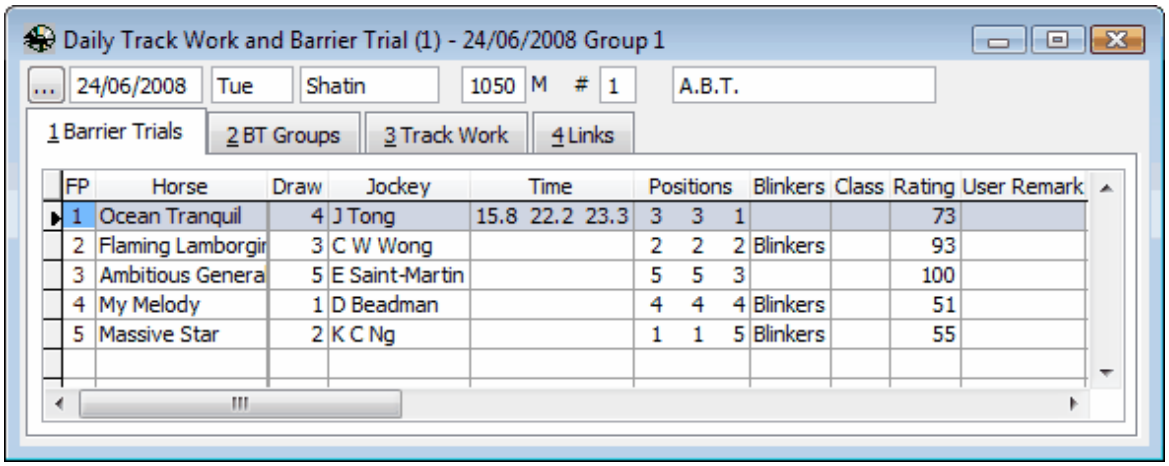
**>>> Subtopics of this section <<<**

© Welkin Systems Limited

- Barrier Trials (Daily Track Work Form)
- Barrier Trial Groups (Daily Track Work Form)
- Track Work (Daily Track Work Form)
- Links Page (Daily Track Work Form)

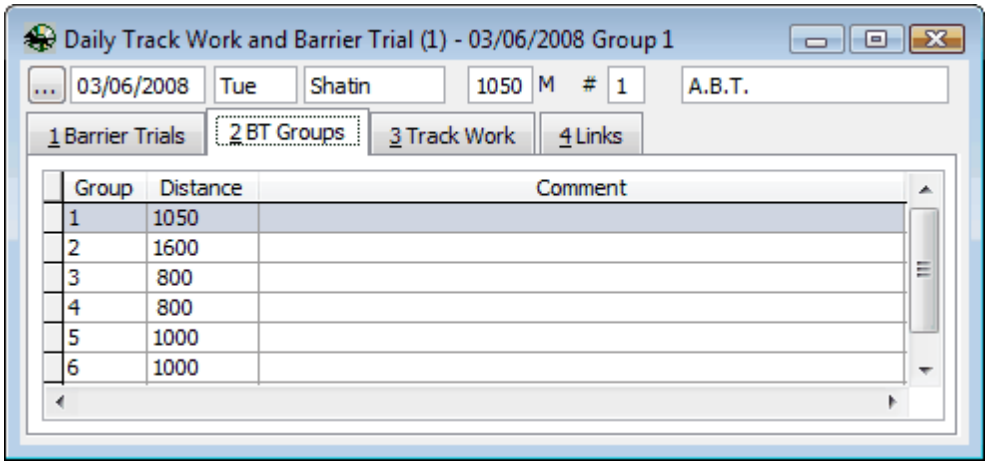
### 5.10.1 Barrier Trials (Daily Track Work Form)

This page show a group of barrier trial horses with related data.


Daily Track Work and Barrier Trial (1) - 24/06/2008 Group 1
24/06/2008
Tue
Shatin
1050
M
#
1
A.B.T.
1 Barrier Trials
2 BT Groups
3 Track Work
4 Links
FP Horse Draw Jockey Time Positions Blinkers Class Rating User Remark
1 Ocean Tranquil 4 J Tong 15.8 22.2 23.3 3 3 1 73
2 Flaming Lamborgir 3 C W Wong 2 2 2 Blinkers 93
3 Ambitious Genera 5 E Saint-Martin 5 5 3 100
4 My Melody 1 D Beadman 4 4 4 Blinkers 51
5 Massive Star 2 K C Ng 1 1 5 Blinkers 55

### 5.10.2 Barrier Trial Groups (Daily Track Work Form)

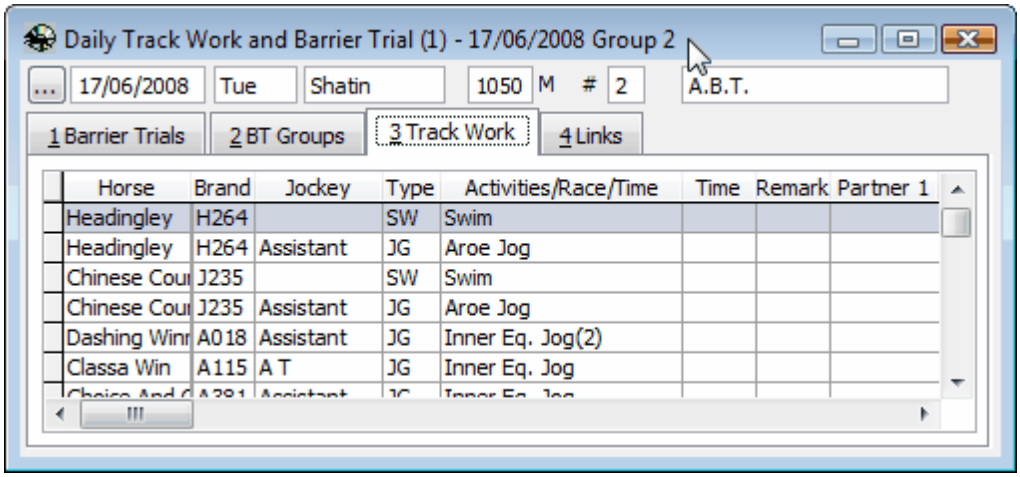
This page displays the barrier trial groups of the day. You may double-click on any of them to show the details of the barrier trial group.




### 5.10.3 Track Work (Daily Track Work Form)

This page show all track work of the day.

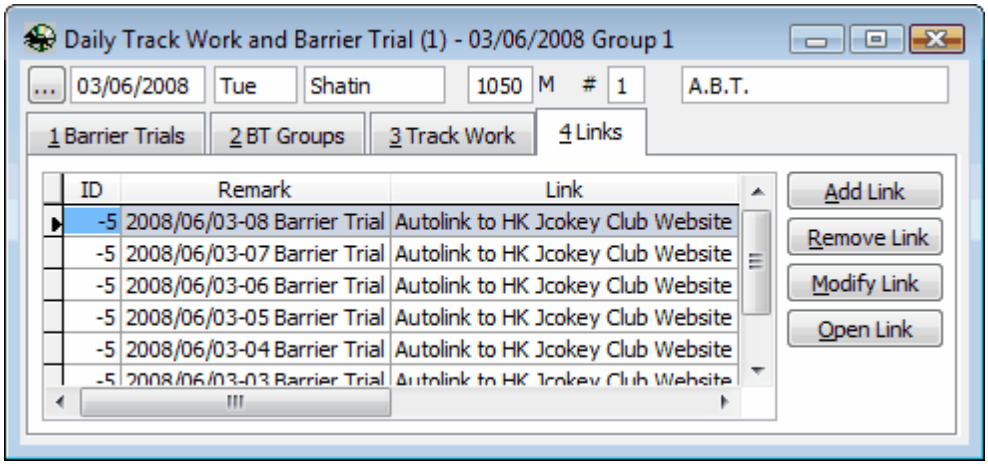
© Welkin Systems Limited




You can input your comments about any track work line into the [Remark] column. After input is finished, push [Ctrl+S] to save the change.

### 5.10.4 Links Page (Daily Track Work Form)

You may open videos of the Jockey Club by double-clicking on the links in the table. The availability of the videos depends on their existence on the server of the Jockey Club. For management about the links, please see the Multimedia Resource section for details.




## 5.11 Jump by Double-Clicking a Column

- Defining Double-Click Actions
- Forms and Record IDs

### 5.11.1 Defining Double-Click Actions

**Type 1: Starter Views**

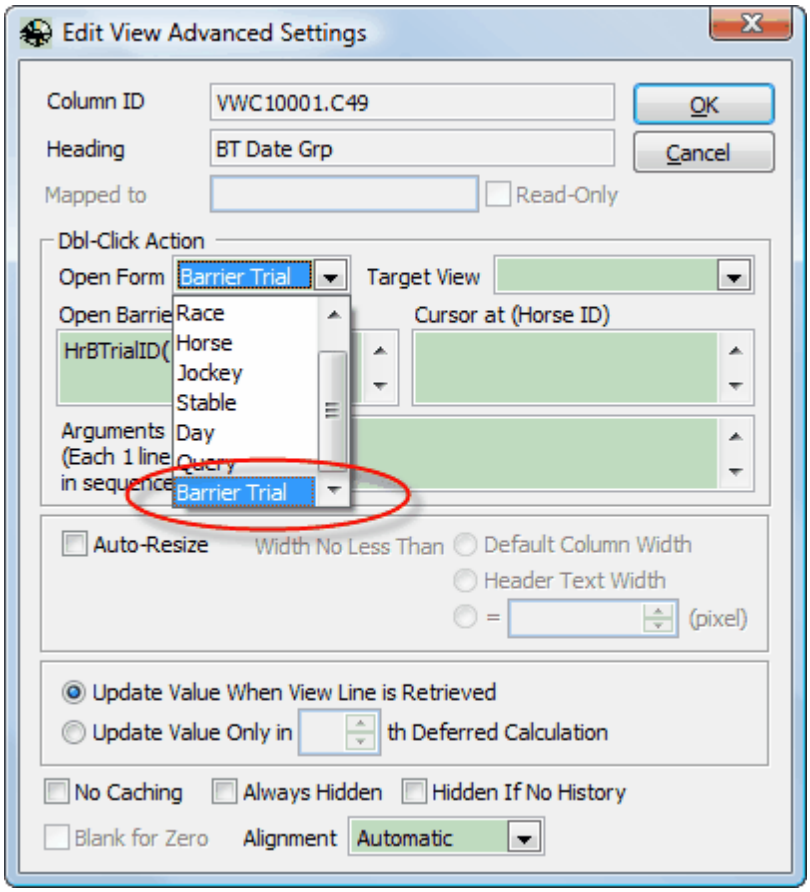
You may follow the steps below to define the response when you double-click on a view column.

1. Click on the [Modify] button on the right-hand side of a Starter View.
2. Select the column you want and click on [Advanced].
3. When the following figure shows, choose a form from the [Open Form] drop down

© Welkin Systems Limited

list.

4. Define an ID for the target form and the ID for jumping to a row. Under normal circumstances, See Forms and Record IDs for details about what these IDs are.

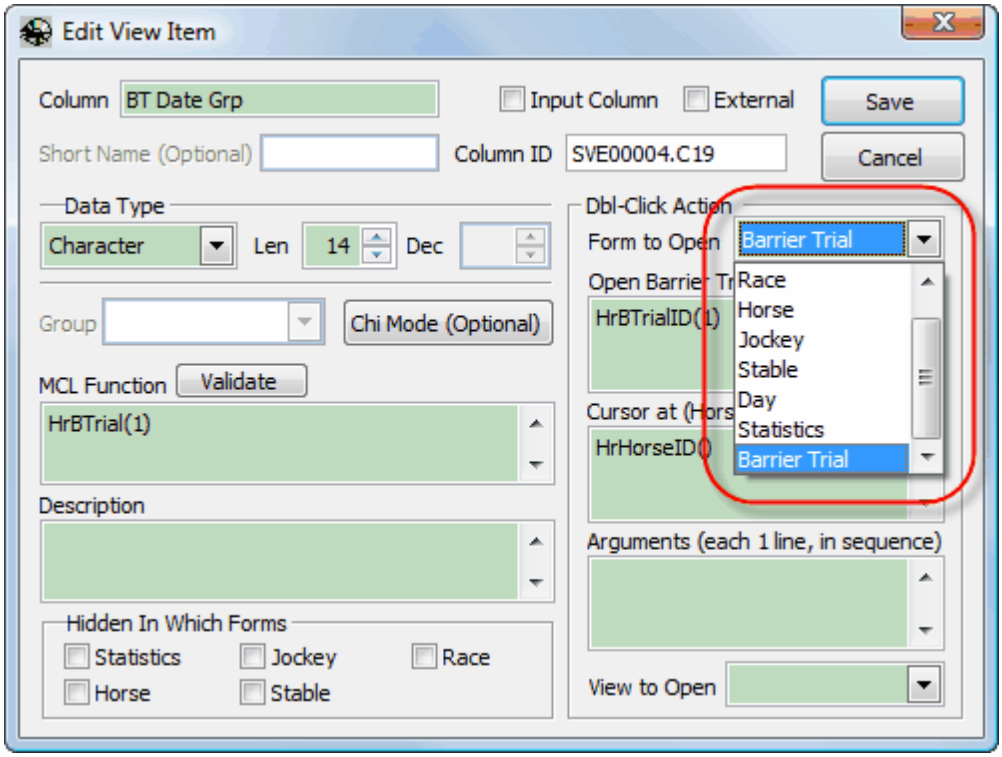



**Type 2: History and Statistics View**

You can follow the steps below to define the target forms for double-click jumping.

1. Click on [Edit View Structure] at the right hand side of the view.
2. Select the column you want to set and click on the [Item Details] button.
3. When the following form shows, choose a target form from the [Form to Open] drop down list.
4. Edit the details about target record and row at the 2 boxes below.

© Welkin Systems Limited




## 5.11.2 Forms and Record IDs

### Forms and Target Record ID

| Form | Record ID | MCL | Example |
|---|---|---|---|
| Race Data | Race ID | HrRaceID( ) | "28346" |
| Horse | Horse ID | HrHorseID( ) | "2XO" |
| Jockey | Jockey ID | HrJockeyID( ) | "2P" |
| Stable | Stable ID | HrStableID( ) | "D0" |
| Day (Calendar) | Race Date | HrDate( ) | {18/03/2009} |
| Barrier Trial | Barrier Trial ID | HrBTrialID( ) | "3514502" |
| Query (Statistics) | / | / | / |

### Row ID

| 表單 | 列的系統代碼 | MCL | 例子 |
|---|---|---|---|
| Race Data | Horse ID | HrHorseID( ) | "2XO" |
| Horse | Race ID | HrRaceID( ) | "28346" |
| Jockey | Race ID | HrRaceID( ) | "28346" |
| Stable | Race ID | HrRaceID( ) | "28346" |
| Barrier Trial | Horse ID | HrHorseID( ) | "2XO" |

# 5.12 Comprehensive Statistics and Record Query Form

See Comprehensive_Statistics_and_Record_Query (under Statistics and Data Analysis) for details.

© Welkin Systems Limited

# Part VI

# 6 Views Management

The following 2 types of views can be found in RaceMate:

1. Starter Views - The views shown on the [Starter Views] page of the Race Data form
2. History and Statistics View - The views on the "Comprehensive Statistics and Record Query" form and the views on the [History/Star] page of the Horse, Jockey, and Stable forms are all History and Statistics View.

See the Statistics and Data Analysis section for more information of the 2 types of views.

This chapter will discussion how to make use of these 2 types of views.

**>>> Subtopics of this section <<<**

- Type 1: Starter Views
- Type 2: History and Statistics Views
- Cache for View Data
- Conversion Tables
- Linking Up Views
- Exporting and Importing View Data

## 6.1 Type 1: Starter Views

Starter Views are views on the [Starter Views] page of the "Race Data" form. On starter views, we can show past record of horses, jockeys or stables by expanding the number of history lines.

A starter view is different from a history/statistics view, which shows on the [History/Stat] page of the Horse, Jockey or Stable forms, in both operations and content. We will discuss the techniques about using this type of view in this chapter.

See the Type 2: Statistics via Starter Views (Race Data Form) section for more information of starter views.

**>>> Subtopics of this section <<<**

- Defining Starter Views
- Show History in Starters Views
- Filtering Records in Starters Views

### 6.1.1 Defining Starter Views

In this section we will discuss how to create and maintain starter views.

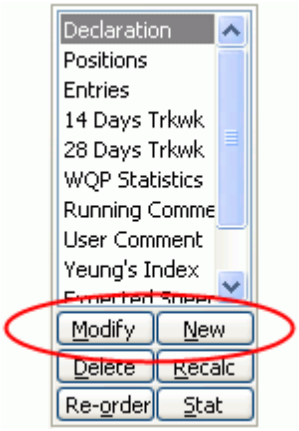
**>>> Subtopics of this section <<<**

© Welkin Systems Limited

- Creating Starter Views
- Modifying Existing Views
- View Header
- Example: Creating a Starter View
- Advanced Settings of View Columns
- Filtering Past Records
- Rearranging Position of Views in View List
- Exporting and Importing View Structures
- Copying a Starter View

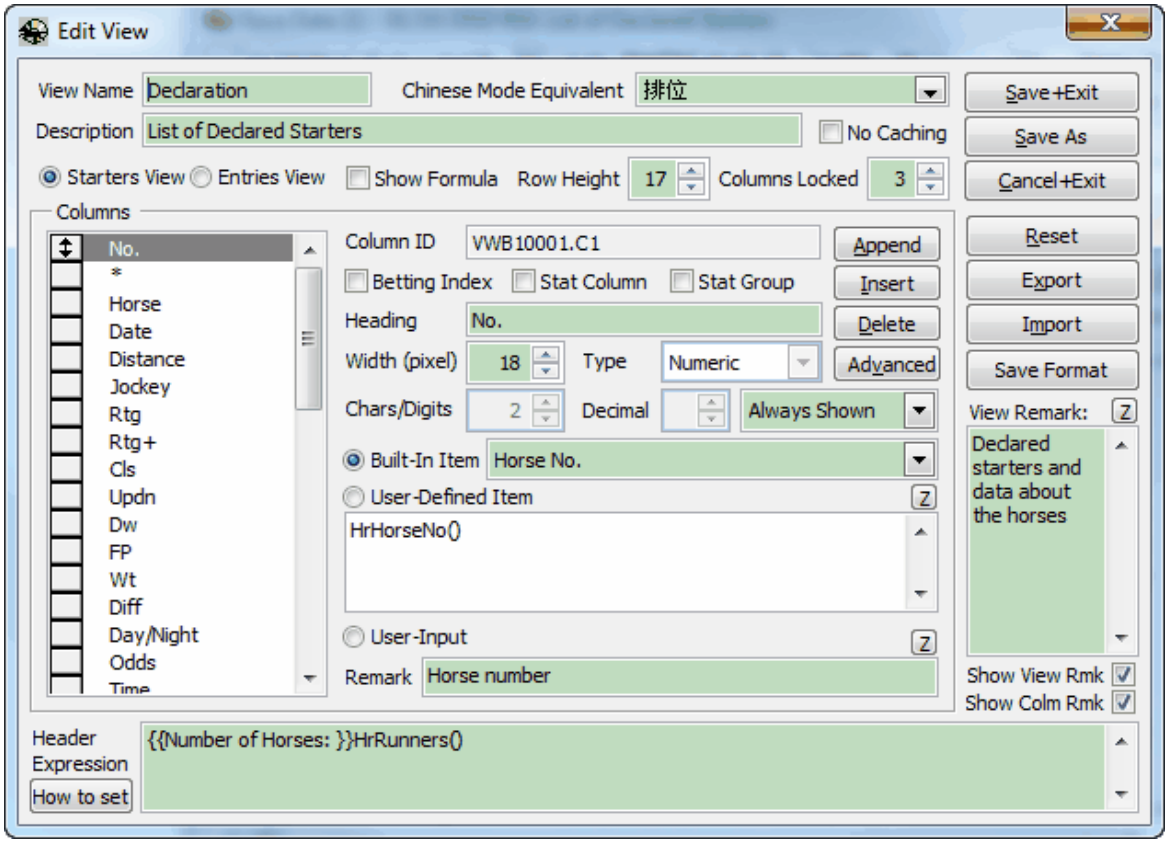
#### 6.1.1.1 Creating Starter Views

To modify the structure of a view, click on the [Modify] button at the right hand side of a view. To add a new view, click on the [New] button.




When you click on the [Modify] or [New] button, RaceMate will show the following "Edit View" form.

© Welkin Systems Limited




1. View Name

   Caption of the view to be shown in the view list

2. Description

   A descriptive long name for the view (up to 80 letters or 40 Chinese characters)

3. Chinese Mode Equivalent

   You may switch between Chinese and English modes in RaceMate Professional Edition. The views of these two modes are independent of one another. The Chinese Mode Equivalent setting determines the Chinese view (the twin view of the current English view) that will get focus when you switch the language mode from English to Chinese.

4. Starters View/Entries View

   Choose between showing starters and entries

5. Show Formula

   When Show Formula is selected, whenever you position your mouse over a view column header, RaceMate will show the formula of the column at the right-top of the screen. It will be helpful to your view designing and testing work. After the view design is completed, you may want to turn it off to make the screen cleaner.

6. Columns Locked

   Set the number of columns which will not be moved with other columns when you scroll the view left and right.

7. Column List

   A list showing all columns of the view. When you select a column in the list, the details of the column will be shown in the controls on the right.

© Welkin Systems Limited

8. Column Reposition Buttons

   You may drag a column reposition button to change the relative position of a column.

9. Column ID

   Each column carries a unique Column ID. You may show the data of a column of another view in the current column by mapping the two columns via the Column ID. (See "Mapped to" below)

10. Heading

    Column caption to be shown at the topic of a column

11. Width

    Column width in pixels

12. Type

    Data type of the column. The following data types are available:

| Data Type | Description |
|---|---|
| Numeric | Integer or decimal numbers. Maximum 20 digits including decimal part and decimal point |
| Character | Text. Maximum 254 characters (127 Chinese characters.) |
| Multiline Text | Block of text. No limited in number of characters. When you increase the row height, a vertical scrollbar will be show at the right edge of the grid cell. Accessing Multiline Text data is slower than accessing Character data. Very often we will use Character type instead of Multiline Text type except the text to be handle is large. |
| Date | Date |
| Logical | True or false. RaceMate will show a u for true and a blank cell for false. If the the logical column is editable, you may click on a cell to change its value from true to false, or vice versa. |

13. Chars/Digits

    Maximum number of characters or digits allowed. A Chinese character will be treated as 2 English letters. For numeric columns, the number includes the decimal part and the dot.

14. Decimal

    Number of digits in the decimal part. Applicable to numeric data only.

15. Display Option Dropdown

    You can use it to choose among the options Always Shown, Always Hidden or Hide if No History.

16. Built-In Item

    RaceMate predefines a set of built-in columns for retrieving data and doing statistics. When you select a built-in item from the built-in item list, the corresponding MCL function will be shown in the expression editing box below. You may convert a Built-In Item Column to a User-Defined Item Column by click on the User-Defined Item button. You may then edit and fine-tune the MCL function.

17. User-Defined Item

    You may enter an MCL expression into the expression editing box to retrieve data or carry out statistics. See the sections of MCL Reference for detailed information about MCL expressions.

© Welkin Systems Limited

18. User-Input Column

    For inputting notes or ratings. A user-input column can be of any of the five data types described above.

19. Remark

    Remark about the column for your own reference

20. Betting Index

    You can mark one (and only one) column as Betting Index. When we carry out an evaluation on the accuracy of the prediction model (i.e., the view), RaceMate will calculate the winning percentage (and investment return) of the horses with highest betting indexes. A view with a betting index defined is called a Prediction Model.

21. Stat Column

    Stat Column is the short form of Statistical Column. When a view is used for statistics, you may mark one (and only one) column as Statistical Column. When statistics is carried out, RaceMate will calculate the following values for the column: Count, Sum, Maximum, Minimum, Average, Variance, and Standard Deviation. We call a view with a statistical column defined a Statistics Model.

22. Stat Groups

    Stat Group is the short form of Statistical Group. You may define unlimited levels of Statistical Groups in a view. RaceMate will find the statistical figures of the statistical column within each group.

23. Append

    Append a new column at the end

24. Insert

    Insert a new column at the current position

25. Delete

    Delete a column

26. Advanced

    See Advanced Settings of View Columns for details

27. Save+Exit

    Save changes and exit to the Race Data Form

28. Cancel+Exit

    Abort changes and exit to the Race Data Form

29. Reset

    Reset the structure of a built-in view to its factory settings. Resetting a view will remove the columns that you added and recall the columns that you deleted.

30. Export

    Export the structure of a view into a disk. The structure can then be read into another RaceMate (of another computer). By using this function, you may transfer the structure of an English view into disk and import the structure into Chinese mode.

31. Import

    Import structure of a view from disk

32. Save Format

© Welkin Systems Limited

In our daily RaceMate operation, we may reorder columns or change column widths of a view using the mouse. Clicking the small button at the left-top of the view will reset the view to its default settings. (The default settings of a user-defined view are the settings in the first save of the column definitions.) Sometimes, you may want the keep your recent changes and don't want them being reset accidentally. Clicking on the [Save Format] button will mark the current settings as the default. Future clicks on the left-top small reset button of a view will put the view into these saved settings.

33. View Remark

    Remarks about the view for your future reference

34. Show View Rmk

    Once this box is checked, when your mouse is positioned above the view name on the view list of the Race Data form, a popup window showing the view remark will be shown.

35. Show Colm Rmk

    Once this box is checked, when your mouse is positioned above the heading of a view column in the Race Data form, a popup showing the remark about the column will be shown.

36. Header Expression

    On the Race Data form, you can show MCL function results and text in the text box above the view table. The Header Expression box is for editing how to show such data. See the View Header page for details.

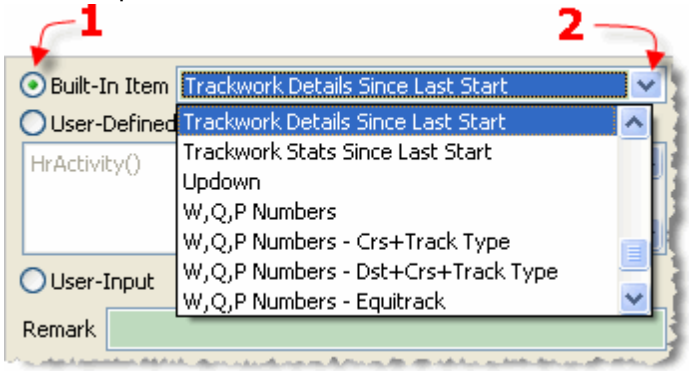
#### 6.1.1.2 Modifying Existing Views

Click on the [Modify] button below the view list to modify a view.

**Adding a New Column**

Click on [Append] to add a new column at the end or [Insert] to add a new column just above the currently selected column.

Built-in Column
A built-in column is a ready made column selectable from the Built-In Item dropdown list. A built-in column comes with predefined format and data type. You may edit its format according to your needs. To insert a built-in column, append or insert a column first. Then click on the [Built-In Item] option button and drop down the item list to select a predefined item.




**User-Defined Column**

A user-defined column is a column with a user-defined expression for retrieving data or

© Welkin Systems Limited

doing statistics. Enter your expression, composed of MCL functions, into the user-defined item expression edit box. The RaceMate online help system provides you with complete information about MCL. Press F1 or click on the (?) button to get access to the help system.

**Context Sensitive MCL Online Help**

While your cursor is staying inside the user-defined item expression edit box, RaceMate provides you with the following shortcuts for getting context sensitive MCL help:

1. If text is selected in the edit box, RaceMate will try to search the selected text in the index of the online help. If an exact match is found, the help page will automatically be opened, otherwise RaceMate will jump to the most closely matched topic.
2. If no text is selected, RaceMate will try to search the help index to match the whole expression in the edit box and jump to the most closely matched topic.

If there is no text in the edit box at all, the "Modify and Adding Views" help page will be displayed.

**Converting a Built-In Column into a User-Defined Column**

If you are not satisfied with the data returned by a built-in column, you may convert a built-in column into a user-defined column by clicking on the User-Defined Column option button. A formula corresponding to the built-in column will automatically be put into the user-defined item edit box. You may then edit the expression according to your need.

**User-Input Column**

Insert a User-Input column if you want to input your personal notes and rating into the view.

**Tips about Disk Space**

Please keep in your mind about the following facts:

1. The larger number of digits/chars assigned to the column, the greater disk space will be consumed. No matter you input how many number of characters into a cell, the same disk space will be used.
2. The "Width (pixel)" you set has no connection with the disk space consumed.
3. You can enter unlimited amount of text into a Multiline Text column. However the space consumed by a multiline column is usually more that other types of columns (significantly more than the actual number of characters entered) and the data retrieval speed is lower than normal columns.

**Deleting a Column**

Click on the [Delete] button to delete a column. If a column is deleted by accident, you may click on the [Cancel+Exit] button to abort all the changes you made to the view.

**Tip:** If you don't want to abort all the changes you made, you may export the current definition by clicking on the [Export] button before aborting. After you have got back your deleted expression, you may than import the view to get back your change.

#### 6.1.1.3 View Header

You can display results of MCL functions in the view header.

**Displaying Text in View Header**

© Welkin Systems Limited

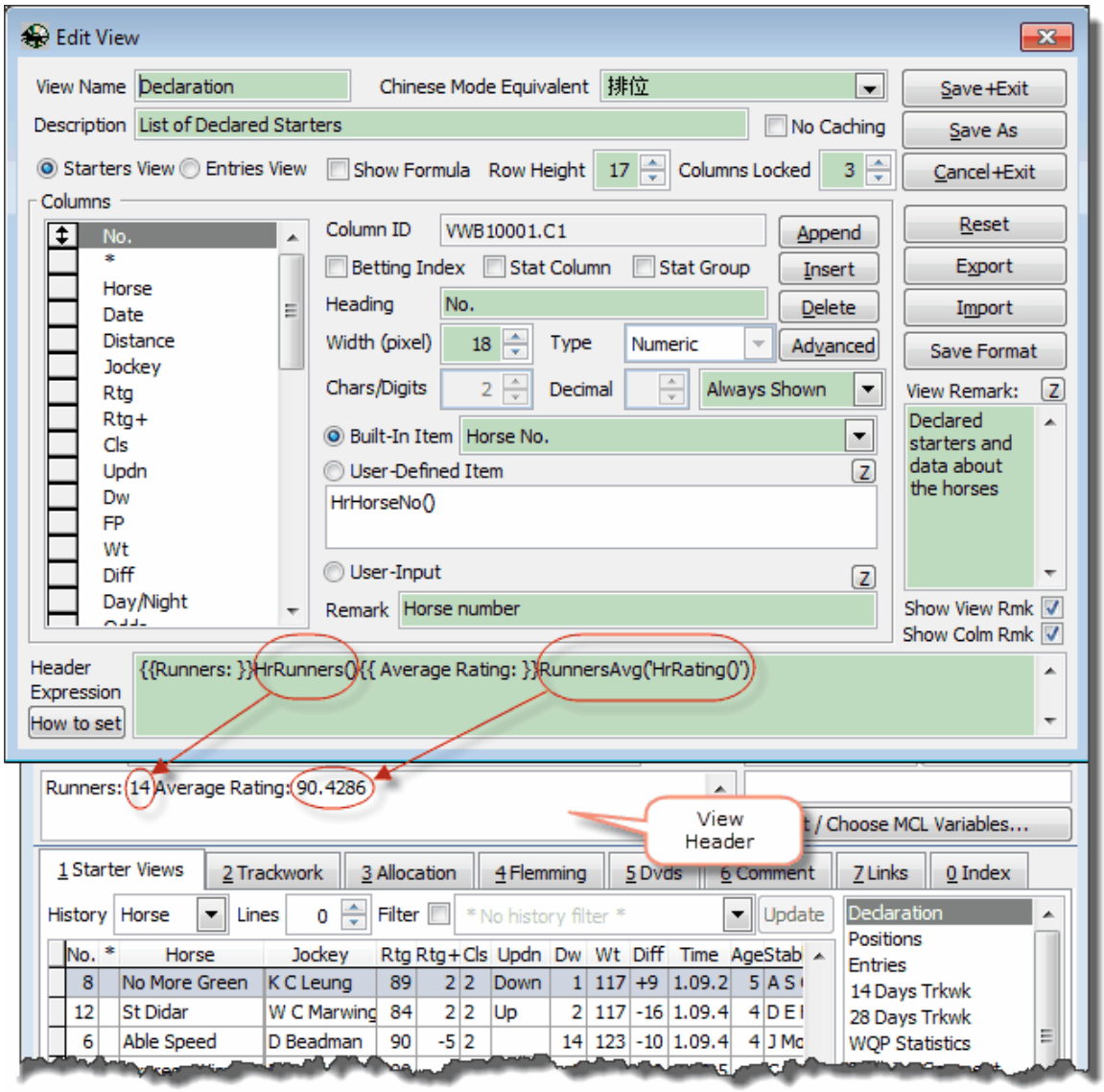
To display text in the view header, put the text inside a pair of {{ and }} quotes into the Header Expression box. We call it a text node. You can put unlimited number of characters inside the quotes. There can be unlimited number of text nodes in a view header.

### Displaying Results of MCL Functions in View Header

To display the result of an MCL function, put the function directly into the Header Expression box. Two MCL functions must be separated by a text node. The simplest text node is the one with no text (only quotes {{}}) or just a single space (i.e., {{ }}).

### Calculation Context for MCL Functions in View Header

MCL functions will be calculated after all columns of the view table below are filled. Calculations will take place in the context of the data for the winning horse (or horse number 1 if the race has not yet be started). Therefore the Column() function will return the view data of the winning horse.



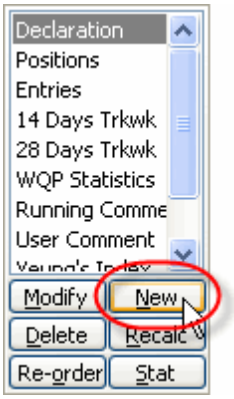

© Welkin Systems Limited

#### 6.1.1.4 Example: Creating a Starter View

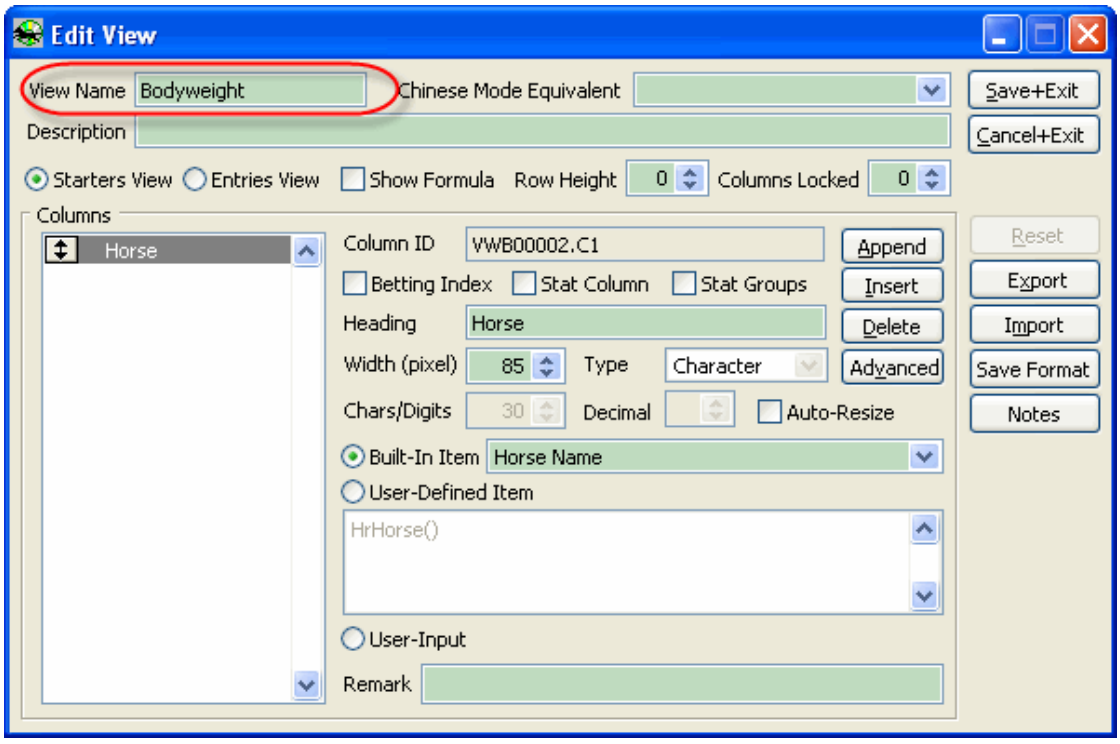
An experimental view about the bodyweight of horses will be created below. Since this is a sample view to demonstrate the basic steps of creating view columns, the formulae are deliberately made simple. If your want to get comprehensive statistics about bodyweight, you will need to create put more columns into the view.

**Creating a View to Retrieve Bodyweight Information**

1. On the Starter Views page of the Race Data Form, click on the [New] button at the right hand side of the page.



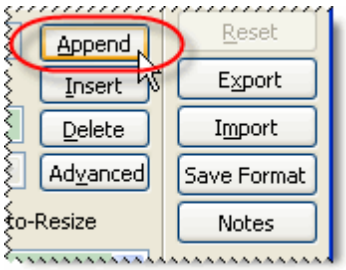

2. You will see an Edit View form. Input "Bodyweight" into the View Name textbox. You will see a Built-In Column "Horse Name" (heading: Horse) is automatically inserted as the first column of the view.



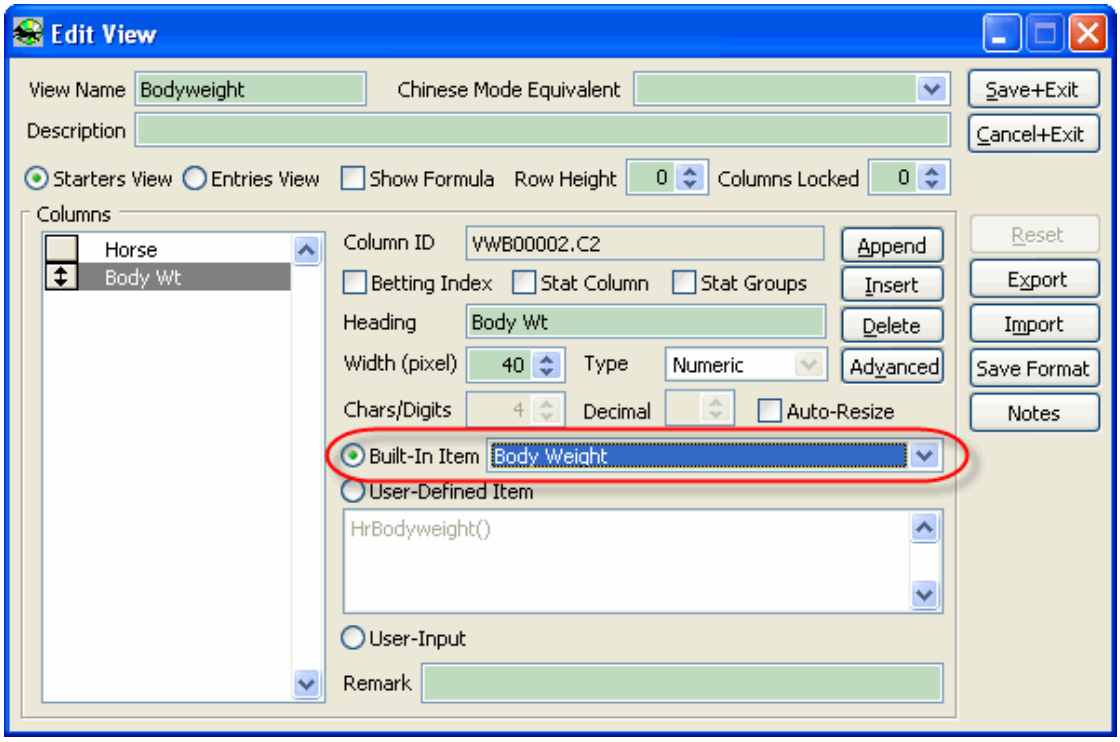

**Add a New Built-In Column**

3. Click on the [Append] button to append a new column

© Welkin Systems Limited




4. Please leave the Heading box blank and choose Built-In Item "Body Weight from the Built-In Item dropdown list.




**Note:**

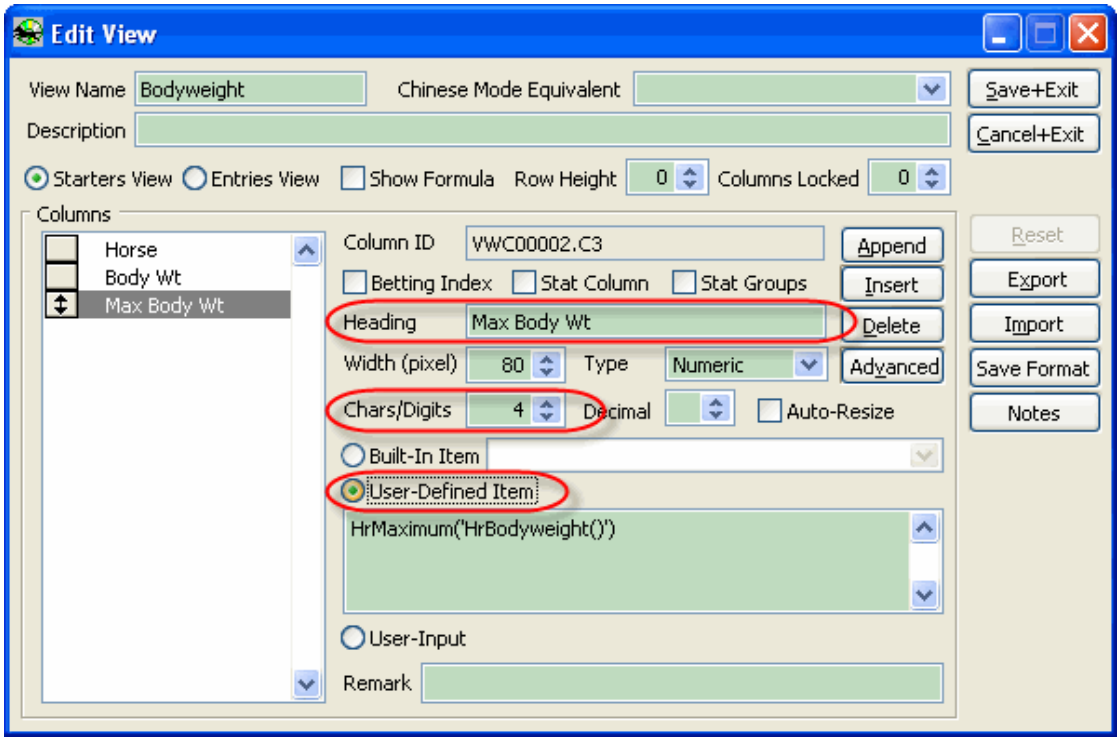
- If the Heading textbox is blank and an item is choose from the Built-In Item dropdown list, the Heading textbox will automatically be filled by RaceMate.
- The MCL function corresponding to the Built-In Item will be shown in the User-Defined Item editbox. If you change the column to a User-Defined Item and use the MCL function for the column, the same result will be shown in the view.

### Create a User-Defined Item

5. Click on the [Append] button again to append a new column. Set the column a User-Define Item with the column name "Max Body Wt".
6. Set Data Type to "Numeric" and Chars/Digits to 4. (The setting will very often be written as Numeric(4) or Numeric(4,0), where "0" means 0 decimal place.
7. Input the following MCL function into the User-Defined Item editbox:

   HrMaximum('HrBodyWeight( )')

© Welkin Systems Limited




### Another User-Defined Item

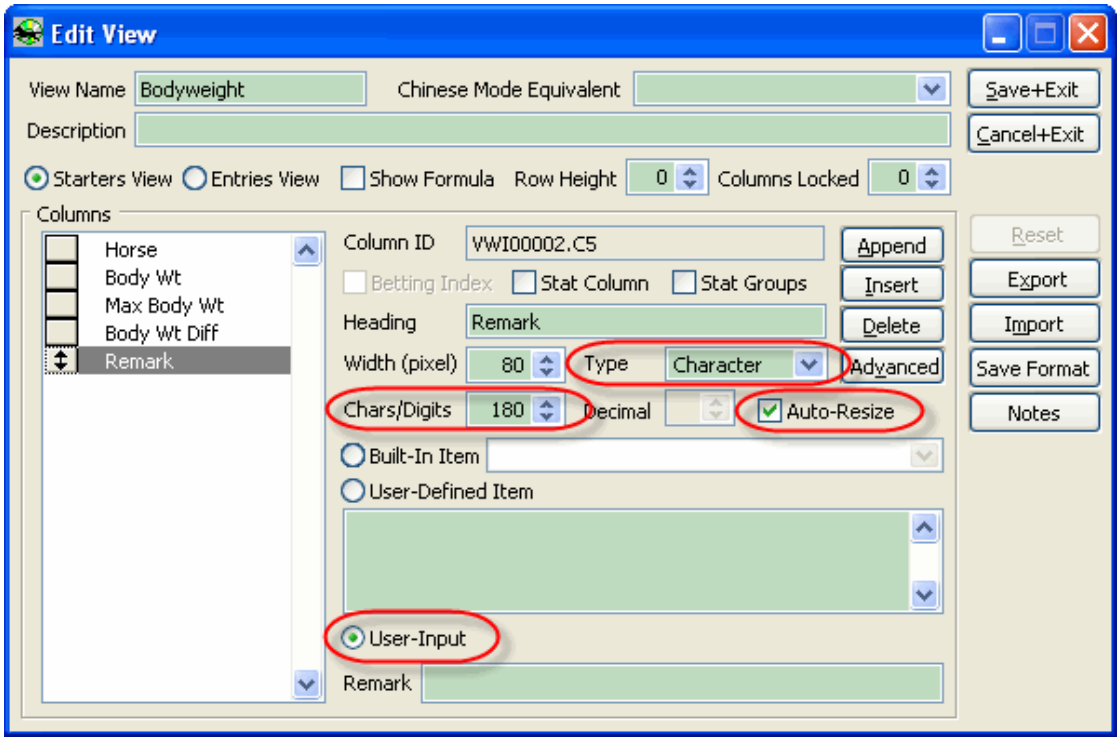
8. Click on [Append] again to add a new column. Chose User-Defined Item and name the Heading "Body Wt Diff".
9. Input the following MCL function into the User-Defined Item edit box:

   Column('Max Body Wt') - Column('Body Wt')

### Create a User-Input Column

10. Click on [Append] to add a new column. Choose "User-Defined Item" and "Character" type.
11. Set number of Chars/Digits to 180 and choose "Auto-Resize" (column width will be changed when you resize the form).

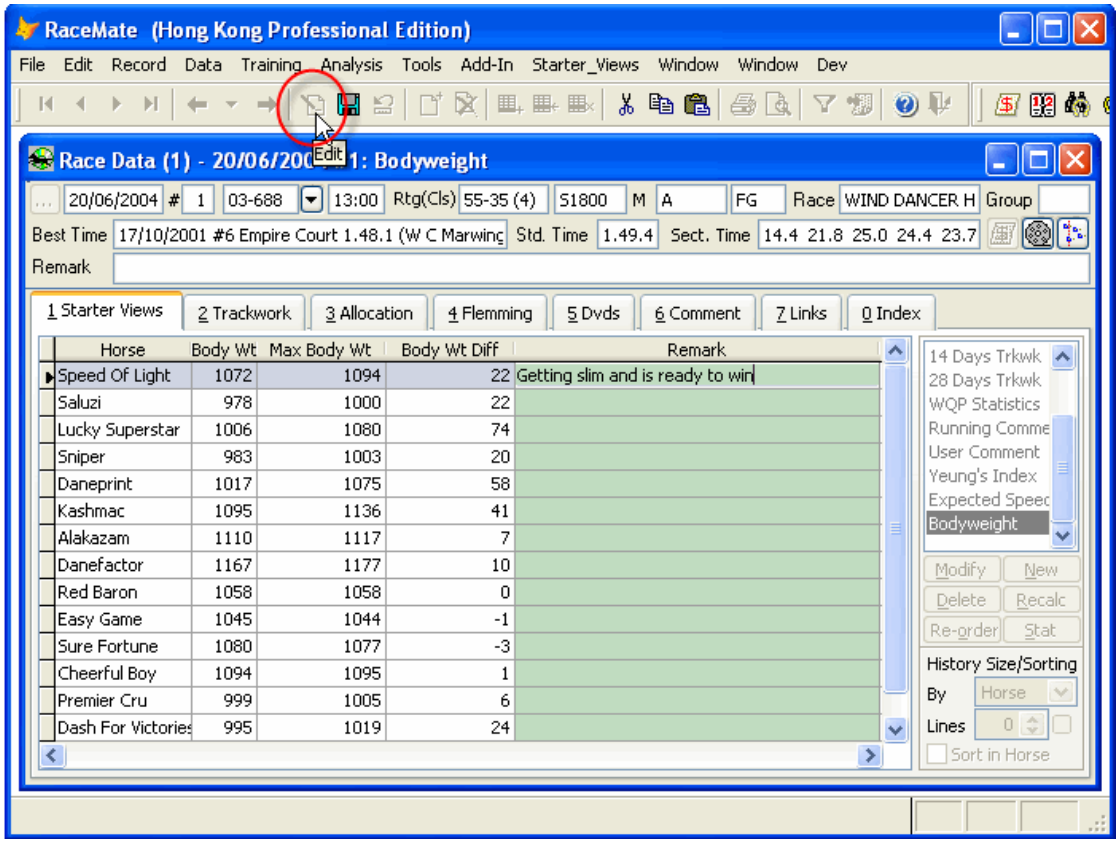
© Welkin Systems Limited




**View Results and Input Remarks**

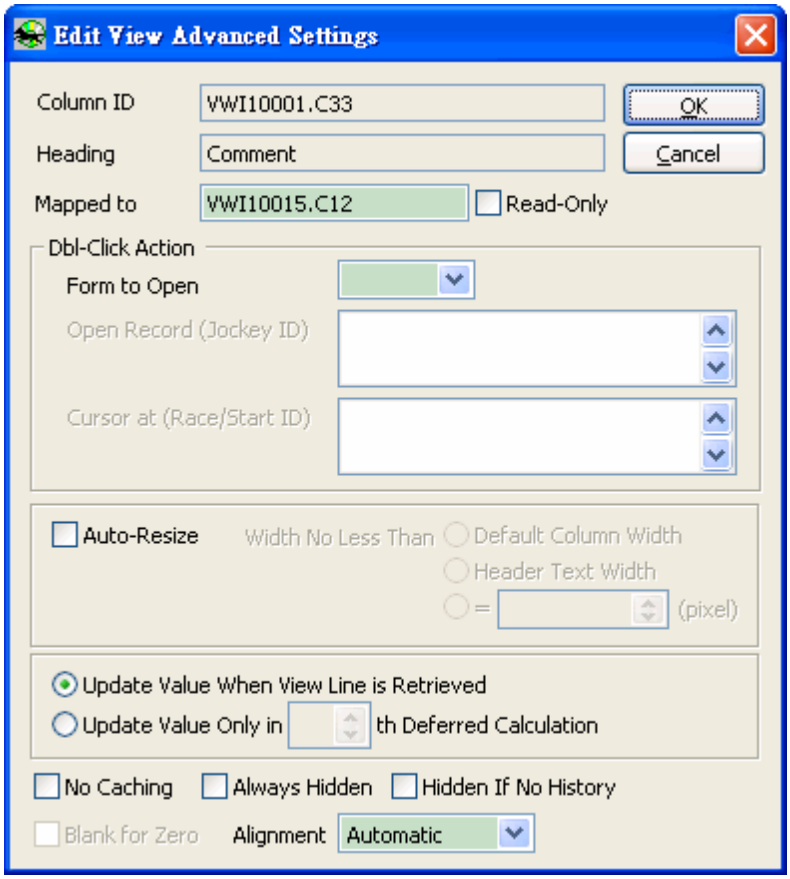
12. Click on "Save+Exit" to save the view and show the view data
13. Click on the [Edit] button of the toolbar to put the form into the "Edit Mode". You will see the color of the Remark column turns green. The color shows that the column is in edit mode and is ready for inputting text.
14. After text input is finished, click on the [Save] button on the toolbar to save the change

© Welkin Systems Limited


RaceMate (Hong Kong Professional Edition)
File Edit Record Data Training Analysis Tools Add-In Starter_Views Window Window Dev
Edit
Race Data (1) - 20/06/20
1: Bodyweight
20/06/2004 # 1 03-688 13:00 Rtg(Cls) 55-35 (4) 51800 M A FG Race WIND DANCER H Group
Best Time 17/10/2001 #6 Empire Court 1.48.1 (W C Marwinc Std. Time 1.49.4 Sect. Time 14.4 21.8 25.0 24.4 23.7
Remark
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Comment
7 Links
0 Index
Horse Body Wt Max Body Wt Body Wt Diff Remark
Speed Of Light 1072 1094 22 Getting slim and is ready to win
Saluzi 978 1000 22
Lucky Superstar 1006 1080 74
Sniper 983 1003 20
Daneprint 1017 1075 58
Kashmac 1095 1136 41
Alakazam 1110 1117 7
Danefactor 1167 1177 10
Red Baron 1058 1058 0
Easy Game 1045 1044 -1
Sure Fortune 1080 1077 -3
Cheerful Boy 1094 1095 1
Premier Cru 999 1005 6
Dash For Victories 995 1019 24
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Expected Speec
Bodyweight
Modify New
Delete Recalc
Re-order Stat
History Size/Sorting
By Horse
Lines 0
Sort in Horse

© Welkin Systems Limited

### 6.1.1.5 Advanced Settings of View Columns




1. Mapped to

   You can display data of a column of another view in the current column by entering the column ID of the target column into this box. Except for special cases, normally we will make the settings (data type, chars/digit, etc.) same as the target column. Otherwise data of the target column may not be properly displayed.

2. Read-Only

   No change made to the mapped column is allowed

3. Form to Open

   Form to be open when the column is double-clicked

4. Open Record

   Specify the record to be open after a form is opened by double-clicking. For example, you can specify here record of which horse will be shown when the Horse form is opened.

5. Cursor at

   Specify the line to be highlighted when a form is opened by double-clicking

6. Auto-Resize

   When checked, the column will automatically be resized with the form. However, resizing takes place only when the total width of all columns is less than the width of the grid. The Auto-Resize columns will expand to fill the blank space of a grid. Contrarily, when the form is shrunk by the user, the width of the column will be reduced automatically. You may mark the minimum width of the column in one of

© Welkin Systems Limited

the following options:

- Default column width – column will always be wider than the width you defined
- Header text width – column can be shrink not thinner than the width of the column caption
- = - column width equals to the pixel number in the spinner

7. No Caching

   RaceMate will save a copy of racing data and calculation results of a view into a cache, so that when the view of the same race is opened next time, calculations of MCL functions will no longer be needed. When this box is checked, RaceMate will always recalculate the MCL function of this column instead of retrieving the value from cache.

8. Always Hidden

   Check this box to hide a column used for interim calculations making the screen cleaner and tidier.

9. Hidden If No History

   The upper part of the Race Data Form shows some race specific data, e.g., Distance, Course, etc. There is no need to repeatedly show such data in each horse row of a view. However, when past records of the horse are displayed, you may want to see such data since such data of the past records are different from that of the current race. Selecting this option will make a column visible only when past record lines are shown.

10. Blank if Zero

    When the value is zero, display nothing rather than 0.

11. Alignment

    Specify how text (or numbers) is aligned in the column.

#### 6.1.1.6 Filtering Past Records

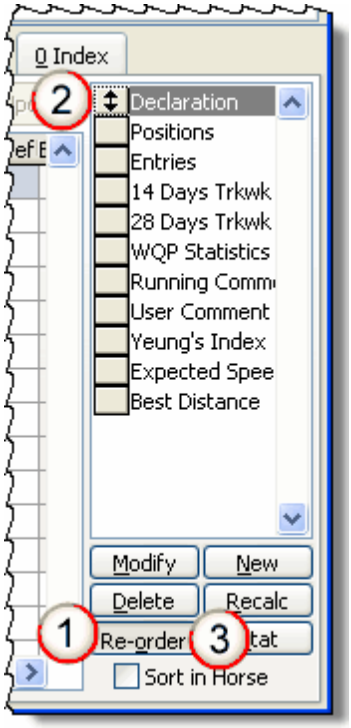
You may show only the records satisfying the criteria that you set. See the section Filtering Records in Starters Views for details.

#### 6.1.1.7 Rearranging Position of Views in View List

Follow the steps below to rearrange the position of the views in the view list:

1. Push down the [Re-order] button (a small button will appear at the left hand side of each view name in the view list)
2. Drag the button of the view that you want to relocate to the new target position
3. Click on the [Re-order] button to release it

© Welkin Systems Limited

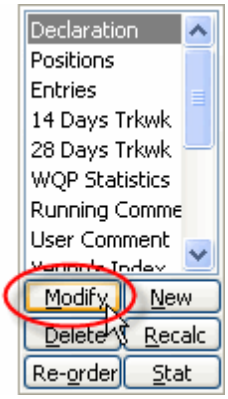



#### 6.1.1.8 Exporting and Importing View Structures

You can export the structure of a starter view into a diskette (or a folder) and then import the view structure into another copy of RaceMate so as to share your view with another RaceMate user. If you carry out the export and import actions in the same RaceMate, the end result is copying the structure of an existing view to a new view.

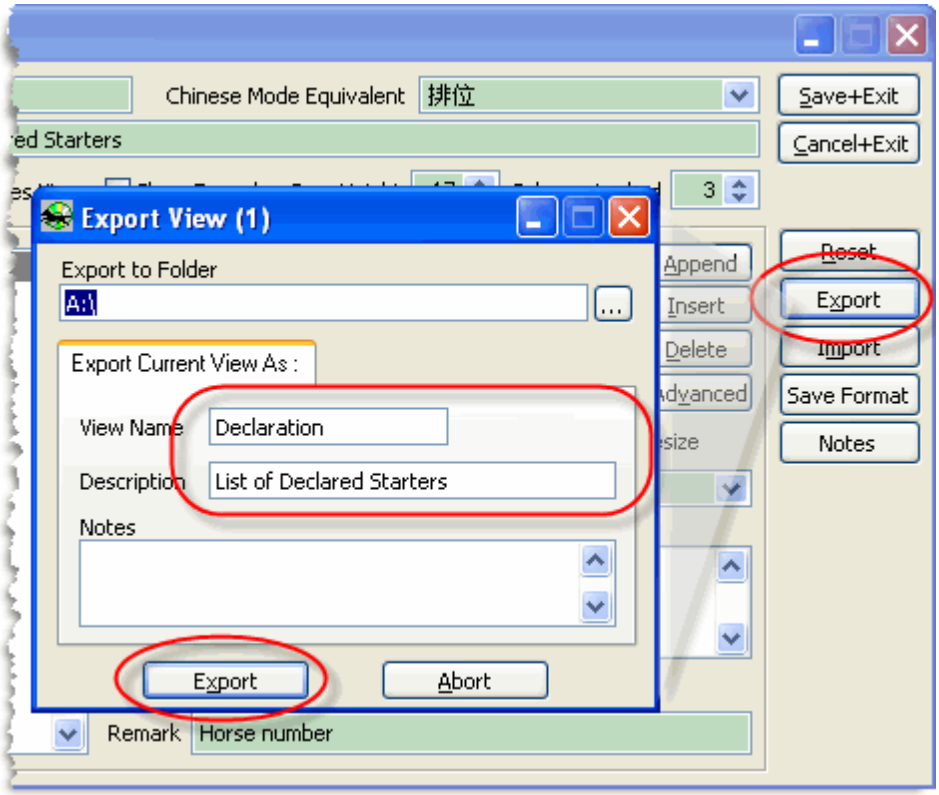
**Steps to export view structure to a disk**

1. On the Race Data form, open the view that you want to export its structure.
2. Click on the [Modify] button at the right hand side of the Race.




3. When the "Edit View" form shows, click on the [Export] button at the right edge of the form.
4. Enter a view name and a description into the Export View form.

© Welkin Systems Limited

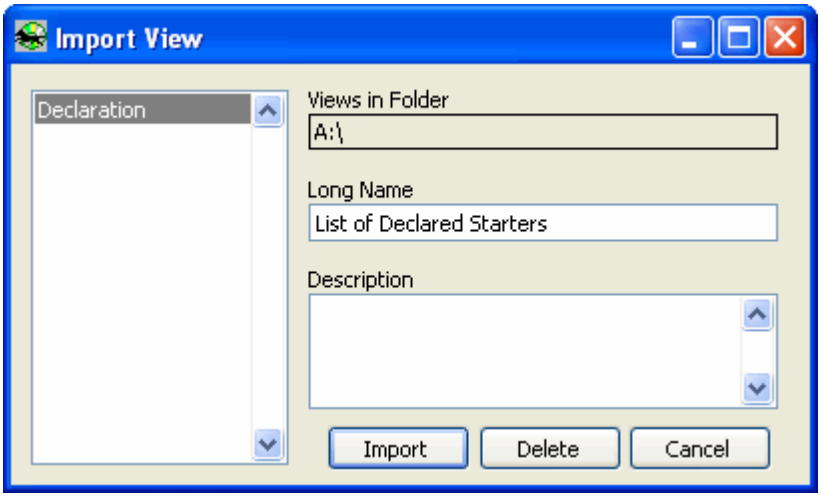



5. Insert a floppy diskette into drive A: to start exporting.

**Steps to import view structure**

1. Start RaceMate in another computer that you want to import view structure.
2. Put the floppy diskette created from the steps above into drive A:.
3. On the Starter View page of the Race Data form, click on the [New] button at the right hand side of the form.
4. When the "Edit View" form shows, click on the import button at the right hand side of the form. A dialog for selecting source folder will be shown.
5. After selecting a folder, you will see the following form (shown below) showing a list of views structures stored in the folder. Select the view structure that your want to import from the list at the left hand side and click on [Import] to start importing.

© Welkin Systems Limited




6. You will see a dialog box telling you that the current view structure being edited will be overwritten by the view structure being imported. In this example, the view being edited is a newly created view. We will answer [Yes] to confirm importing.

#### 6.1.1.9 Copying a Starter View

Steps to copy a view definition

1. Export the view to be copied
2. Create a blank new view
3. Import the previously exported view
4. Give a new name to the imported view

Please note that the data that you input in the original view will not be copied to the new view.

### 6.1.2 Show History in Starters Views

**>>> Subtopics of this section <<<**

- History Tracks
- Sorting and Grouping
- Sorting in Individual History Groups
- Setting Values & Choosing MCL Variables

#### 6.1.2.1 History Tracks

On the Race Data form, you can show history of each horse by choosing "Horse" on the "History" track drop down list and set number of history lines. The following screen capture shows 3 past records of each horse.

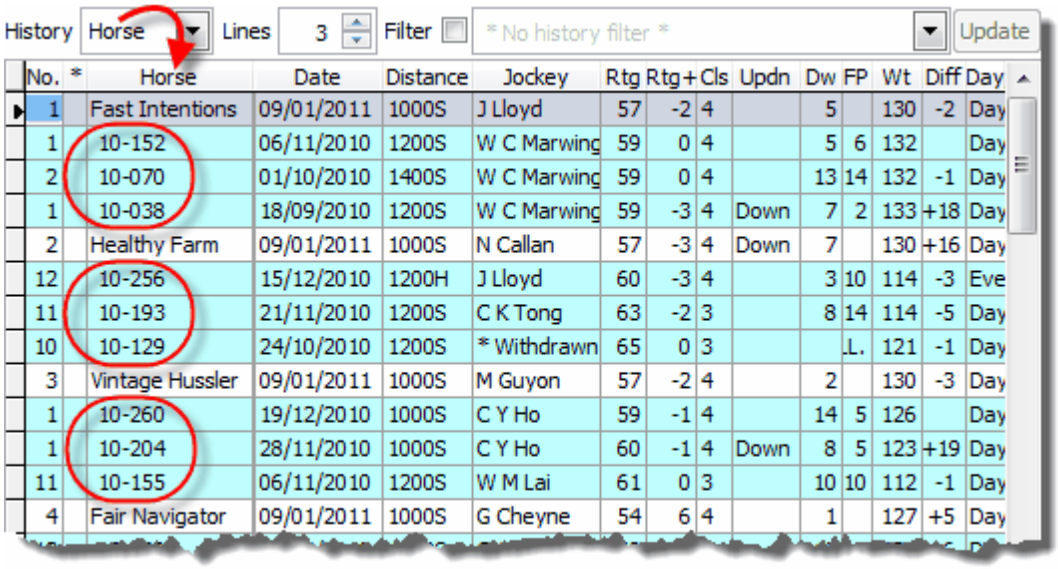
There are 3 history tracks in total:

- Horse
- Jockey
- Stable

© Welkin Systems Limited

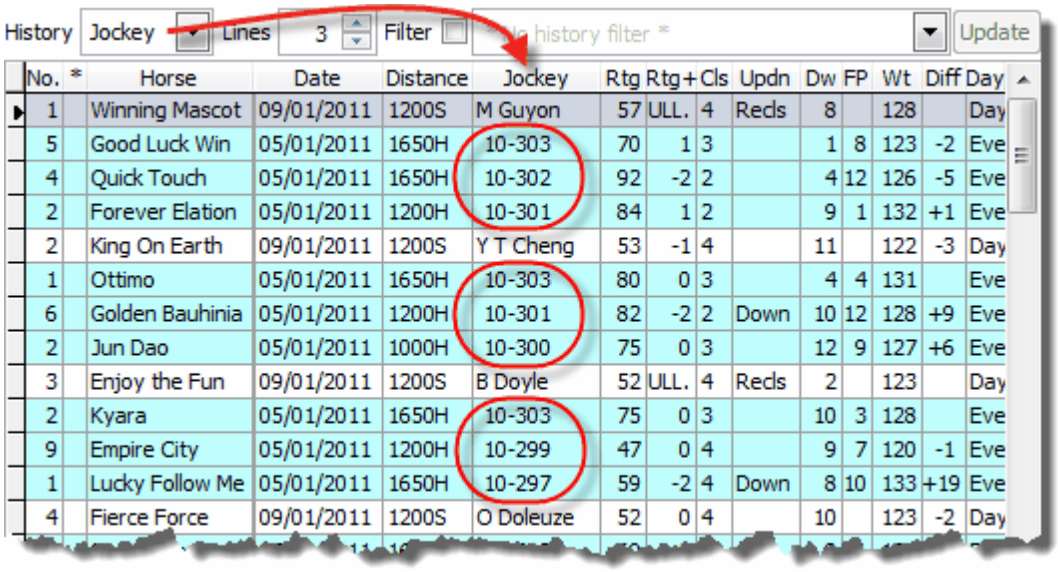
When the form shows, it will keep the history track that you last chosen.

1. Horse history track: 3 latest horse start records.




| No. | * | Horse | Date | Distance | Jockey | Rtg | Rtg+ | Cls | Updn | Dw | FP | Wt | Diff | Day |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Fast Intentions | 09/01/2011 | 1000S | J Lloyd | 57 | -2 | 4 | | 5 | | 130 | -2 | Day |
| 1 | | 10-152 | 06/11/2010 | 1200S | W C Marwing | 59 | 0 | 4 | | 5 | 6 | 132 | | Day |
| 2 | | 10-070 | 01/10/2010 | 1400S | W C Marwing | 59 | 0 | 4 | | 13 | 14 | 132 | -1 | Day |
| 1 | | 10-038 | 18/09/2010 | 1200S | W C Marwing | 59 | -3 | 4 | Down | 7 | 2 | 133 | +18 | Day |
| 2 | | Healthy Farm | 09/01/2011 | 1000S | N Callan | 57 | -3 | 4 | Down | 7 | | 130 | +16 | Day |
| 12 | | 10-256 | 15/12/2010 | 1200H | J Lloyd | 60 | -3 | 4 | | 3 | 10 | 114 | -3 | Eve |
| 11 | | 10-193 | 21/11/2010 | 1200S | C K Tong | 63 | -2 | 3 | | 8 | 14 | 114 | -5 | Day |
| 10 | | 10-129 | 24/10/2010 | 1200S | * Withdrawn | 65 | 0 | 3 | | | LL. | 121 | -1 | Day |
| 3 | | Vintage Hussler | 09/01/2011 | 1000S | M Guyon | 57 | -2 | 4 | | 2 | | 130 | -3 | Day |
| 1 | | 10-260 | 19/12/2010 | 1000S | C Y Ho | 59 | -1 | 4 | | 14 | 5 | 126 | | Day |
| 1 | | 10-204 | 28/11/2010 | 1000S | C Y Ho | 60 | -1 | 4 | Down | 8 | 5 | 123 | +19 | Day |
| 11 | | 10-155 | 06/11/2010 | 1200S | W M Lai | 61 | 0 | 3 | | 10 | 10 | 112 | -1 | Day |
| 4 | | Fair Navigator | 09/01/2011 | 1000S | G Cheyne | 54 | 6 | 4 | | 1 | | 127 | +5 | Day |

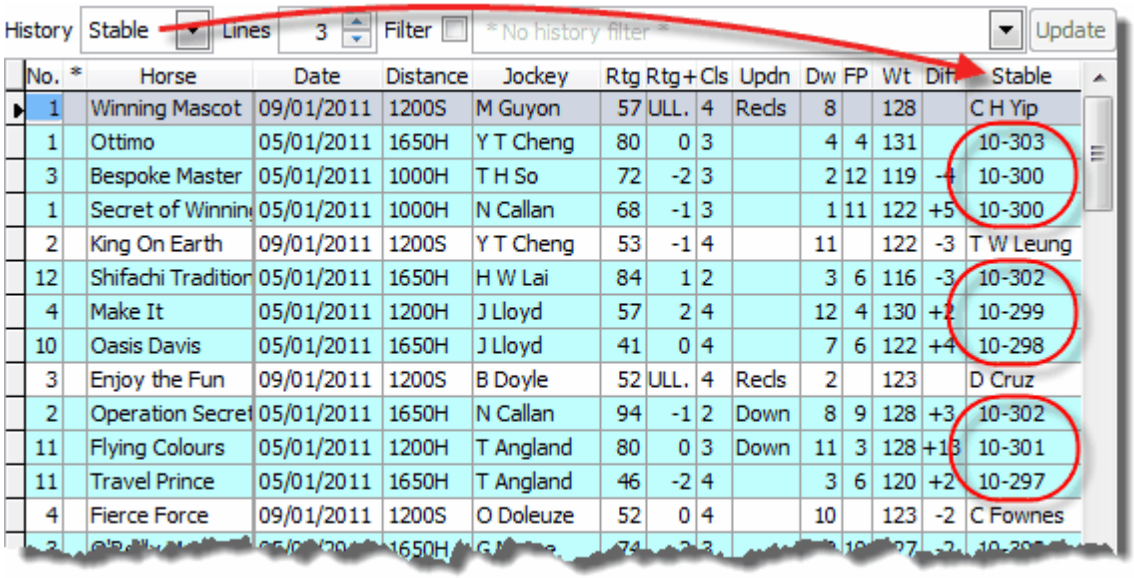
2. Jockey history track: 3 latest jockey records.




| No. | * | Horse | Date | Distance | Jockey | Rtg | Rtg+ | Cls | Updn | Dw | FP | Wt | Diff | Day |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Winning Mascot | 09/01/2011 | 1200S | M Guyon | 57 | ULL. | 4 | Recls | 8 | | 128 | | Day |
| 5 | | Good Luck Win | 05/01/2011 | 1650H | 10-303 | 70 | 1 | 3 | | 1 | 8 | 123 | -2 | Eve |
| 4 | | Quick Touch | 05/01/2011 | 1650H | 10-302 | 92 | -2 | 2 | | 4 | 12 | 126 | -5 | Eve |
| 2 | | Forever Elation | 05/01/2011 | 1200H | 10-301 | 84 | 1 | 2 | | 9 | 1 | 132 | +1 | Eve |
| 2 | | King On Earth | 09/01/2011 | 1200S | Y T Cheng | 53 | -1 | 4 | | 11 | | 122 | -3 | Day |
| 1 | | Ottimo | 05/01/2011 | 1650H | 10-303 | 80 | 0 | 3 | | 4 | 4 | 131 | | Eve |
| 6 | | Golden Bauhinia | 05/01/2011 | 1200H | 10-301 | 82 | -2 | 2 | Down | 10 | 12 | 128 | +9 | Eve |
| 2 | | Jun Dao | 05/01/2011 | 1000H | 10-300 | 75 | 0 | 3 | | 12 | 9 | 127 | +6 | Eve |
| 3 | | Enjoy the Fun | 09/01/2011 | 1200S | B Doyle | 52 | ULL. | 4 | Recls | 2 | | 123 | | Day |
| 2 | | Kyara | 05/01/2011 | 1650H | 10-303 | 75 | 0 | 3 | | 10 | 3 | 128 | | Eve |
| 9 | | Empire City | 05/01/2011 | 1200H | 10-299 | 47 | 0 | 4 | | 9 | 7 | 120 | -1 | Eve |
| 1 | | Lucky Follow Me | 05/01/2011 | 1650H | 10-297 | 59 | -2 | 4 | Down | 8 | 10 | 133 | +19 | Eve |
| 4 | | Fierce Force | 09/01/2011 | 1200S | O Doleuze | 52 | 0 | 4 | | 10 | | 123 | -2 | Day |

2. Stable history track: 3 latest stable records.

© Welkin Systems Limited


History Stable Lines 3 Filter * No history filter * Update
No. * Horse Date Distance Jockey Rtg Rtg+ Cls Updn Dw FP Wt Diff Stable
1 Winning Mascot 09/01/2011 1200S M Guyon 57 ULL. 4 Recls 8 128 C H Yip
1 Ottimo 05/01/2011 1650H Y T Cheng 80 0 3 4 4 131 10-303
3 Bespoke Master 05/01/2011 1000H T H So 72 -2 3 2 12 119 -4 10-300
1 Secret of Winnin 05/01/2011 1000H N Callan 68 -1 3 1 11 122 +5 10-300
2 King On Earth 09/01/2011 1200S Y T Cheng 53 -1 4 11 122 -3 T W Leung
12 Shifachi Tradition 05/01/2011 1650H H W Lai 84 1 2 3 6 116 -3 10-302
4 Make It 05/01/2011 1200H J Lloyd 57 2 4 12 4 130 +2 10-299
10 Oasis Davis 05/01/2011 1650H J Lloyd 41 0 4 7 6 122 +4 10-298
3 Enjoy the Fun 09/01/2011 1200S B Doyle 52 ULL. 4 Recls 2 123 D Cruz
2 Operation Secret 05/01/2011 1650H N Callan 94 -1 2 Down 8 9 128 +3 10-302
11 Flying Colours 05/01/2011 1200H T Angland 80 0 3 Down 11 3 128 +13 10-301
11 Travel Prince 05/01/2011 1650H T Angland 46 -2 4 3 6 120 +2 10-297
4 Fierce Force 09/01/2011 1200S O Doleuze 52 0 4 10 123 -2 C Fownes

### 6.1.2.2 Sorting and Grouping

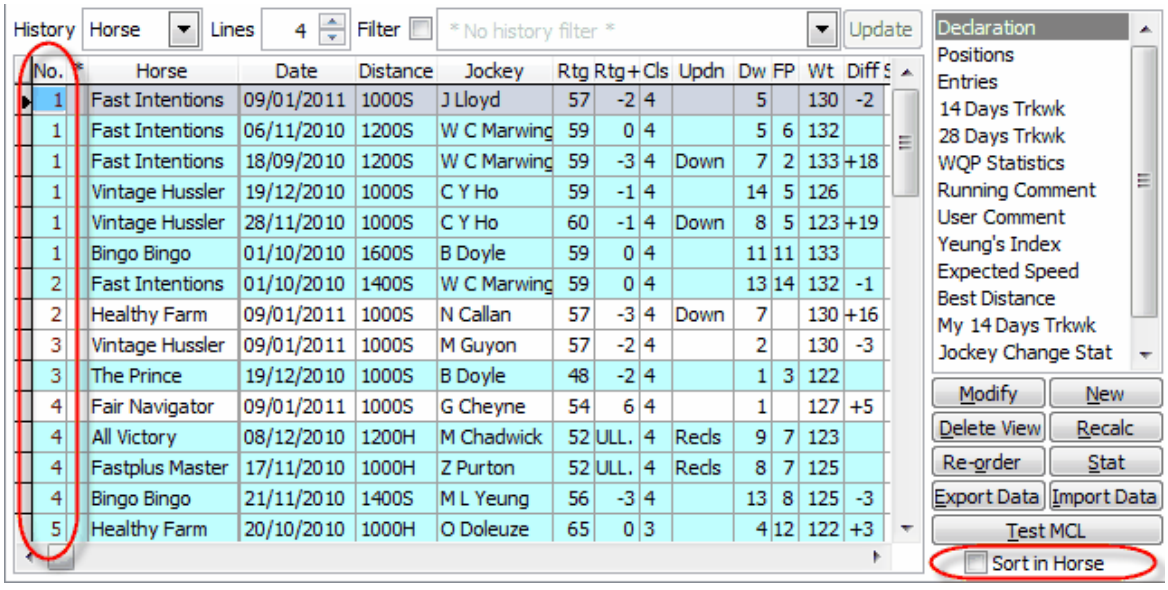
No matter history lines have been shown or not, you can may sort view lines by clicking on a column heading. For techniques of sorting, please see the page Sorting Grid Columns.

### 6.1.2.3 Sorting in Individual History Groups

When history lines are shown, there are 2 different sorting modes:

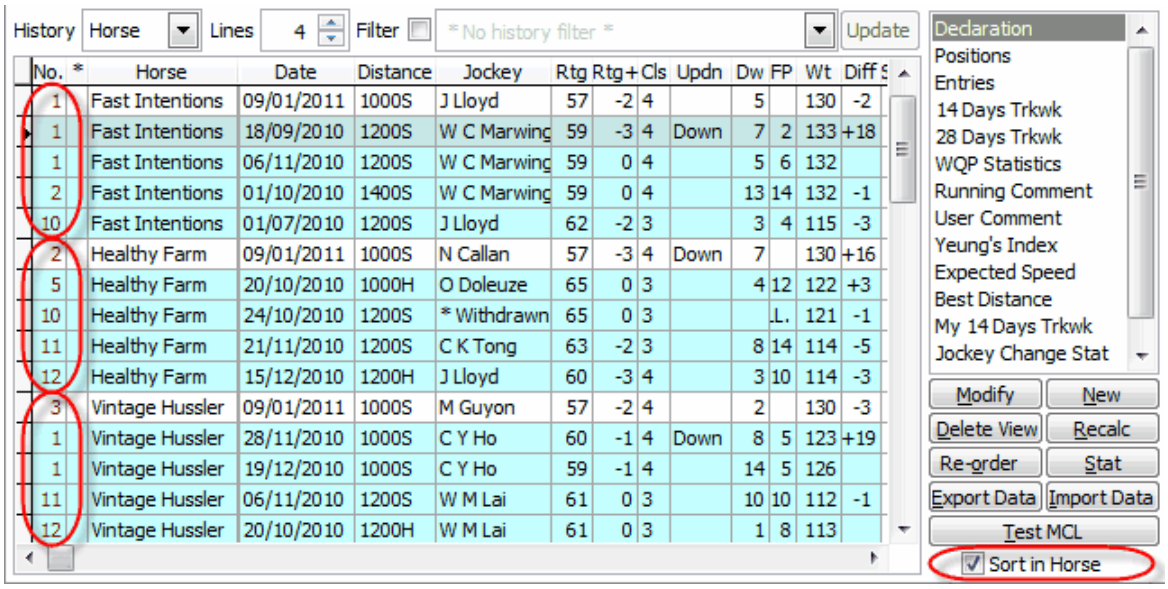
1. All lines sort
2. Sort in horse

The following screen capture shows an "all line sort": keep the [Sort in Horse] check box deselected and click on the column heading. All lines in the view are treated as a group in sorting.


History Horse Lines 4 Filter * No history filter * Update
No. * Horse Date Distance Jockey Rtg Rtg+ Cls Updn Dw FP Wt Diff
1 Fast Intentions 09/01/2011 1000S J Lloyd 57 -2 4 5 130 -2
1 Fast Intentions 06/11/2010 1200S W C Marwing 59 0 4 5 6 132
1 Fast Intentions 18/09/2010 1200S W C Marwing 59 -3 4 Down 7 2 133 +18
1 Vintage Hussler 19/12/2010 1000S C Y Ho 59 -1 4 14 5 126
1 Vintage Hussler 28/11/2010 1000S C Y Ho 60 -1 4 Down 8 5 123 +19
1 Bingo Bingo 01/10/2010 1600S B Doyle 59 0 4 11 11 133
2 Fast Intentions 01/10/2010 1400S W C Marwing 59 0 4 13 14 132 -1
2 Healthy Farm 09/01/2011 1000S N Callan 57 -3 4 Down 7 130 +16
3 Vintage Hussler 09/01/2011 1000S M Guyon 57 -2 4 2 130 -3
3 The Prince 19/12/2010 1000S B Doyle 48 -2 4 1 3 122
4 Fair Navigator 09/01/2011 1000S G Cheyne 54 6 4 1 127 +5
4 All Victory 08/12/2010 1200H M Chadwick 52 ULL. 4 Recls 9 7 123
4 Fastplus Master 17/11/2010 1000H Z Purton 52 ULL. 4 Recls 8 7 125
4 Bingo Bingo 21/11/2010 1400S M L Yeung 56 -3 4 13 8 125 -3
5 Healthy Farm 20/10/2010 1000H O Doleuze 65 0 3 4 12 122 +3
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
User Comment
Yeung's Index
Expected Speed
Best Distance
My 14 Days Trkwk
Jockey Change Stat
Modify New
Delete View Recalc
Re-order Stat
Export Data Import Data
Test MCL
Sort in Horse

© Welkin Systems Limited

The following screen capture shows a "sort in horse": keep the [Sort in Horse] check box selected and click on the column heading. Records of each horse are treated as a single group in sorting.


History
Horse
Lines
4
Filter
* No history filter *
Update
No. * Horse Date Distance Jockey Rtg Rtg+ Cls Updn Dw FP Wt Diff
1 Fast Intentions 09/01/2011 1000S J Lloyd 57 -2 4 5 130 -2
1 Fast Intentions 18/09/2010 1200S W C Marwing 59 -3 4 Down 7 2 133 +18
1 Fast Intentions 06/11/2010 1200S W C Marwing 59 0 4 5 6 132
2 Fast Intentions 01/10/2010 1400S W C Marwing 59 0 4 13 14 132 -1
10 Fast Intentions 01/07/2010 1200S J Lloyd 62 -2 3 3 4 115 -3
2 Healthy Farm 09/01/2011 1000S N Callan 57 -3 4 Down 7 130 +16
5 Healthy Farm 20/10/2010 1000H O Doleuze 65 0 3 4 12 122 +3
10 Healthy Farm 24/10/2010 1200S * Withdrawn 65 0 3 L. 121 -1
11 Healthy Farm 21/11/2010 1200S C K Tong 63 -2 3 8 14 114 -5
12 Healthy Farm 15/12/2010 1200H J Lloyd 60 -3 4 3 10 114 -3
3 Vintage Hussler 09/01/2011 1000S M Guyon 57 -2 4 2 130 -3
1 Vintage Hussler 28/11/2010 1000S C Y Ho 60 -1 4 Down 8 5 123 +19
1 Vintage Hussler 19/12/2010 1000S C Y Ho 59 -1 4 14 5 126
11 Vintage Hussler 06/11/2010 1200S W M Lai 61 0 3 10 10 112 -1
12 Vintage Hussler 20/10/2010 1200H W M Lai 61 0 3 1 8 113
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
User Comment
Yeung's Index
Expected Speed
Best Distance
My 14 Days Trkwk
Jockey Change Stat
Modify
New
Delete View
Recalc
Re-order
Stat
Export Data
Import Data
Test MCL
Sort in Horse

#### 6.1.2.4 Setting Values & Choosing MCL Variables

You can embed variables in MCL functions of a view. When the values of the MCL variables change, MCL functions will retrieve different data. See Setting Values & Choosing MCL Variables to learn how to do it.

### 6.1.3 Filtering Records in Starters Views

**>>> Subtopics of this section <<<**

- Selecting Criteria
- Modifying Criteria and Setting Compound Filters
- Filter Editor

#### 6.1.3.1 Selecting Criteria

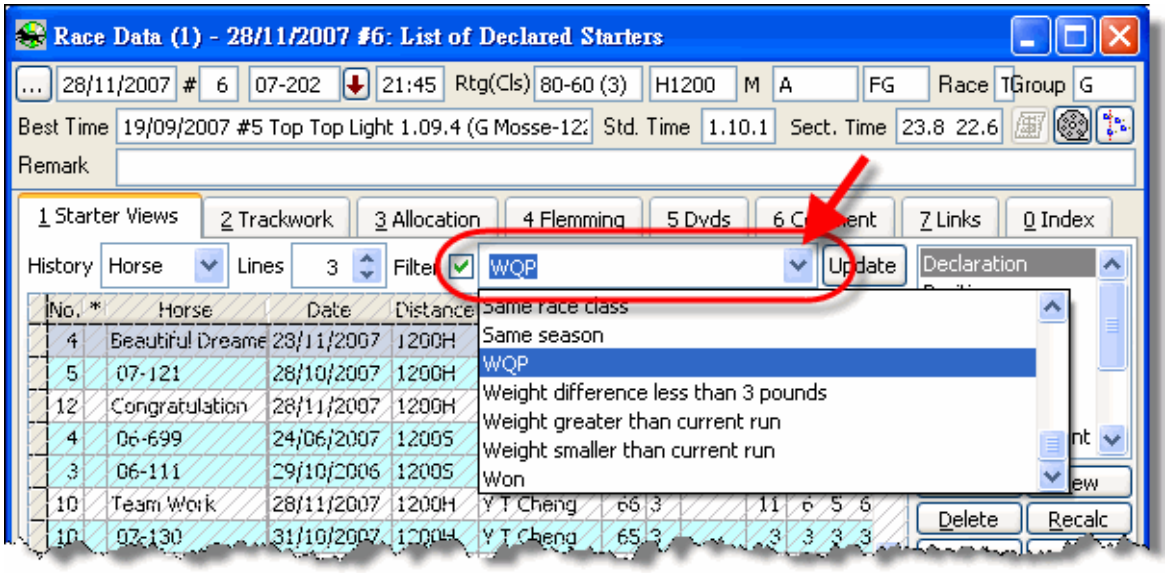
**Setting Filters**

Past records can be shown on a Starter View by choosing a record track and number of record lines at the left top of the Starter View page. You may set one or more than one past record filter and ask RaceMate to show only the past records satisfying the filter(s). The steps are as follows.

**Choosing a Predefined Filter**

You may choose a predefined filter from the filter dropdown. Records shown below the starting horse row are the run(s) satisfying the filtering criterion that you selected. Other runs (not satisfying the filter) are skipped.

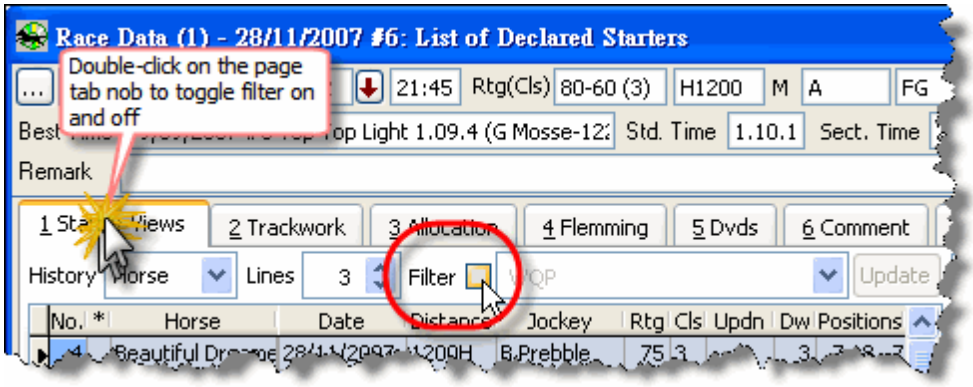
© Welkin Systems Limited


Race Data (1) - 28/11/2007 #6: List of Declared Starters
28/11/2007 # 6 07-202 21:45 Rtg(Cls) 80-60 (3) H1200 M A FG Race TGroup G
Best Time 19/09/2007 #5 Top Top Light 1.09.4 (G Mosse-12: Std. Time 1.10.1 Sect. Time 23.8 22.6
Remark
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
7 Links
0 Index
History Horse Lines 3 Filter WQP Update Declaration
No. * Horse Date Distance
4 Beautiful Dreame 23/11/2007 1200H
5 07-121 28/10/2007 1200H
12 Congratulation 28/11/2007 1200H
4 06-699 24/06/2007 1200S
3 06-111 29/10/2006 1200S
10 Team Work 28/11/2007 1200H Y T Cheng 66 3
10 07-130 31/10/2007 1200H Y T Cheng 65 3
Same race class
Same season
WQP
Weight difference less than 3 pounds
Weight greater than current run
Weight smaller than current run
Won
Delete
Recalc

### Turning Filters Off / On

You may turn filters off and on via one of the following ways:

1. Check/uncheck the [Filter] checkbox at the left hand side of the filter dropdown. After you have cleared the checkbox, the filter definition will still be kept in the editor. You may turn the filter(s) on once again by checking the box.
2. Double-click on the label of the Starter View page to toggle filters on and off


Race Data (1) - 28/11/2007 #6: List of Declared Starters
Double-click on the page tab nob to toggle filter on and off
21:45 Rtg(Cls) 80-60 (3) H1200 M A FG
Std. Time 1.10.1 Sect. Time
Remark
2 Trackwork
4 Flemming
5 Dvds
6 Comment
History Lines 3 Filter Update
No. * Horse Date Jockey Rtg Cls Updn Dw Positions
B.Prebble 75 3

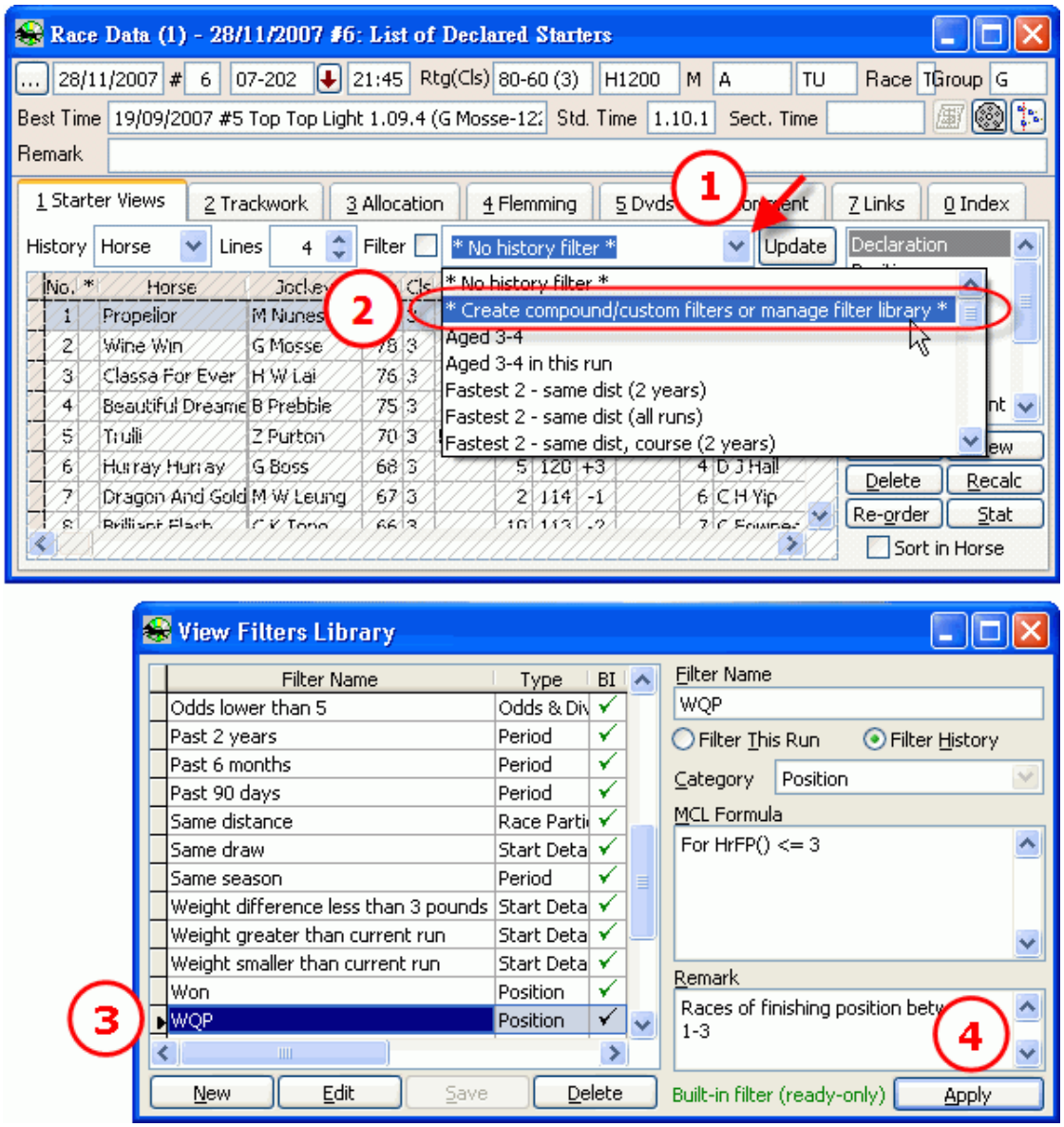
## 6.1.3.2 Modifying Criteria and Setting Compound Filters

### Setting Multiple Filters

1. Display the "Filters Editor" form by carrying out one of the following actions:
   - Select [Create compound/custom filters or manage filter library] from the filter list.
   - Click the [Filter] button of the RaceMate toolbar. (You may also select [Filter] from the [Record] menu. RaceMate will show a Filters Editor form.
2. On the Filters Editor form, click on the [Select from Library] button to show the View Filters Library form.
3. On the left pane of the View Filters Library form, select the filter that you want to apply. (In the following example, filter "WQP" is selected.)

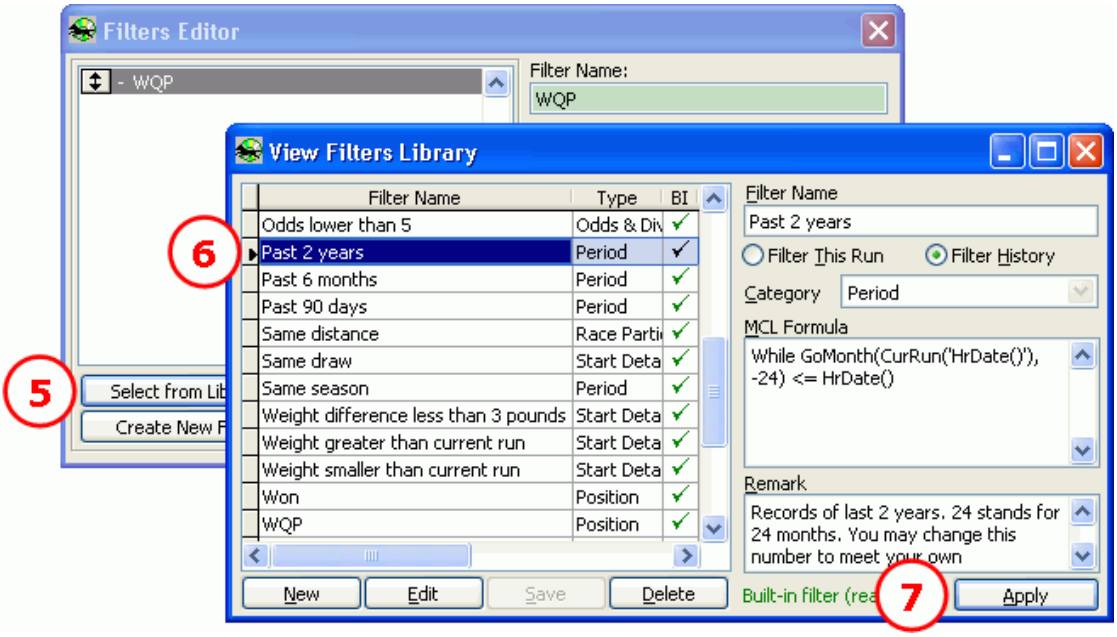
© Welkin Systems Limited

4. Click on the [Apply] button to add the selected filter into the Filters Editor form. We may add more than one filter into the Filter Editor form (see step below).



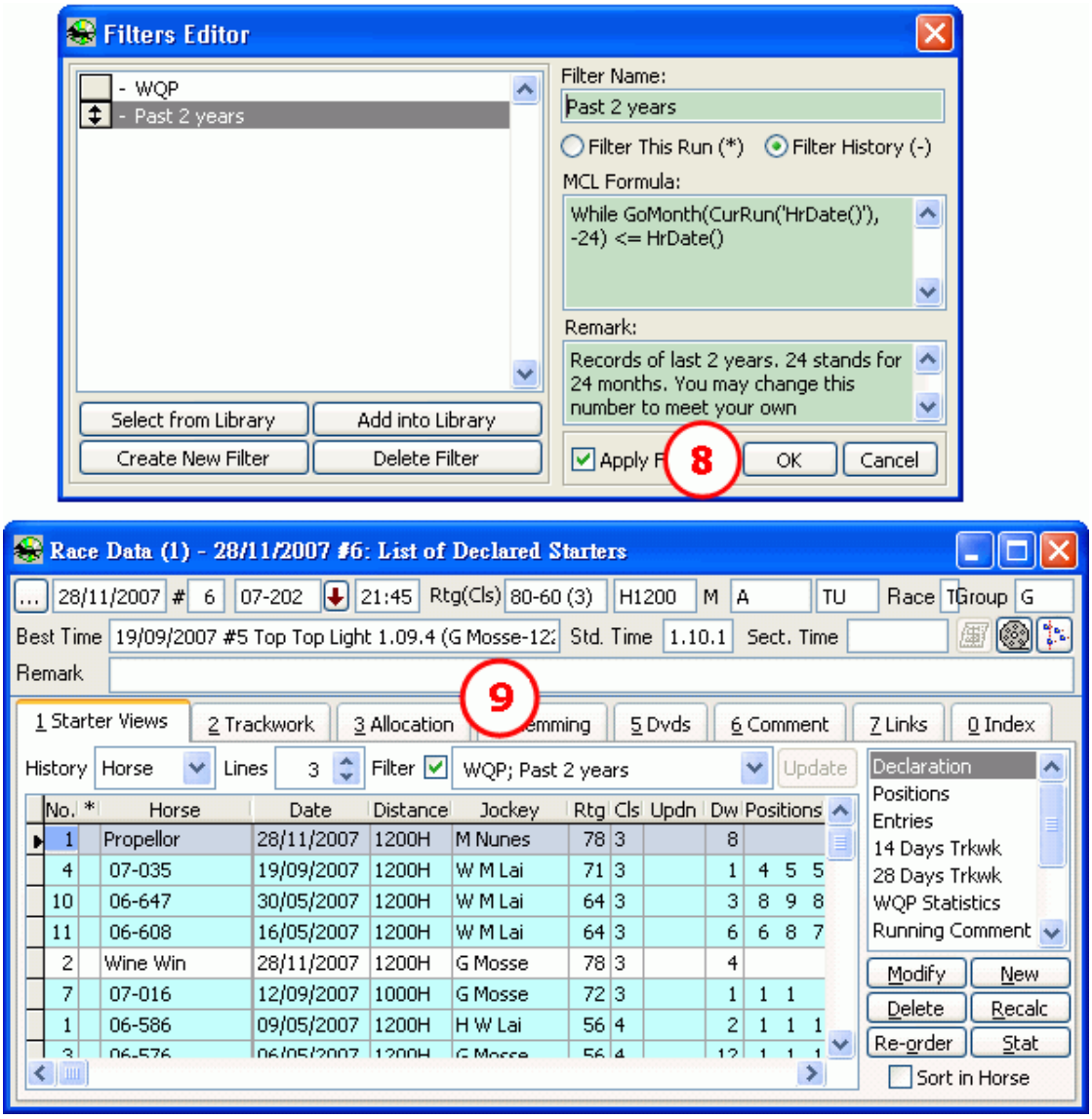

5. If you want to continue to add another filter, you may click on the [Select from Library] button on the Filters Editor form once again to show the View Filters Library form.
6. Select the filter that you want to add as the second filter. (In the following example, "Past 2 years" is selected.)
7. Click on the [Apply] button to add the filter into the Filters Editor.

© Welkin Systems Limited




8. After all filters have been selected. Click on the [OK] button of the Filters Editor form.

© Welkin Systems Limited


Filters Editor
- WQP
- Past 2 years
Filter Name:
Past 2 years
Filter This Run (*)
Filter History (-)
MCL Formula:
While GoMonth(CurRun('HrDate()'), -24) <= HrDate()
Remark:
Records of last 2 years. 24 stands for 24 months. You may change this number to meet your own
Select from Library
Add into Library
Create New Filter
Delete Filter
Apply F
8
OK
Cancel
Race Data (1) - 28/11/2007 #6: List of Declared Starters
28/11/2007 # 6 07-202 21:45 Rtg(Cls) 80-60 (3) H1200 M A TU Race TGroup G
Best Time 19/09/2007 #5 Top Top Light 1.09.4 (G Mosse-12: Std. Time 1.10.1 Sect. Time
Remark
9
1 Starter Views
2 Trackwork
3 Allocation
5 Dvds
6 Comment
7 Links
0 Index
History Horse Lines 3 Filter WQP; Past 2 years Update
No. * Horse Date Distance Jockey Rtg Cls Updn Dw Positions
1 Propellor 28/11/2007 1200H M Nunes 78 3 8
4 07-035 19/09/2007 1200H W M Lai 71 3 1 4 5 5
10 06-647 30/05/2007 1200H W M Lai 64 3 3 8 9 8
11 06-608 16/05/2007 1200H W M Lai 64 3 6 6 8 7
2 Wine Win 28/11/2007 1200H G Mosse 78 3 4
7 07-016 12/09/2007 1000H G Mosse 72 3 1 1 1
1 06-586 09/05/2007 1200H H W Lai 56 4 2 1 1 1
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
Modify
New
Delete
Recalc
Re-order
Stat
Sort in Horse

9. After filters are defined, the past records show on the Starters View will be limited to the runs which satisfy the filters. You can change the number of past record lines. However, some horse may not have the number of starts that meets all the filter criteria you set. (In this example Lighting The World has only one race that passes the filters.)

### 6.1.3.3 Filter Editor

#### Showing the Filters

Follow the steps shown in previous section, open the Filters Editor to modify the filter definitions, add or remove filters. You may open the Filter Editor via one of the following ways:

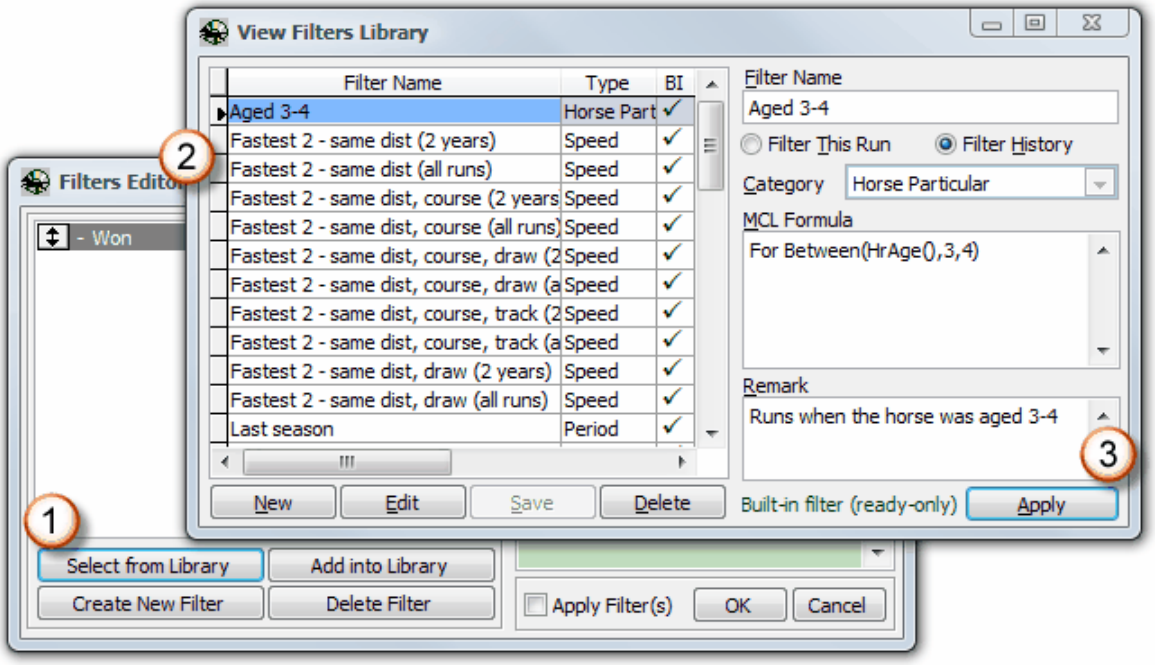
1. Click on the [Filter] button of the RaceMate toolbar, or
2. Choose [Filter] from the [Record] menu, or
3. Select [Create compound/custom filters or manage filter library] from the filter list.

#### Selecting Predefined Filters

On the Filter Editor click on the [Select from Library] button to show the View Filters

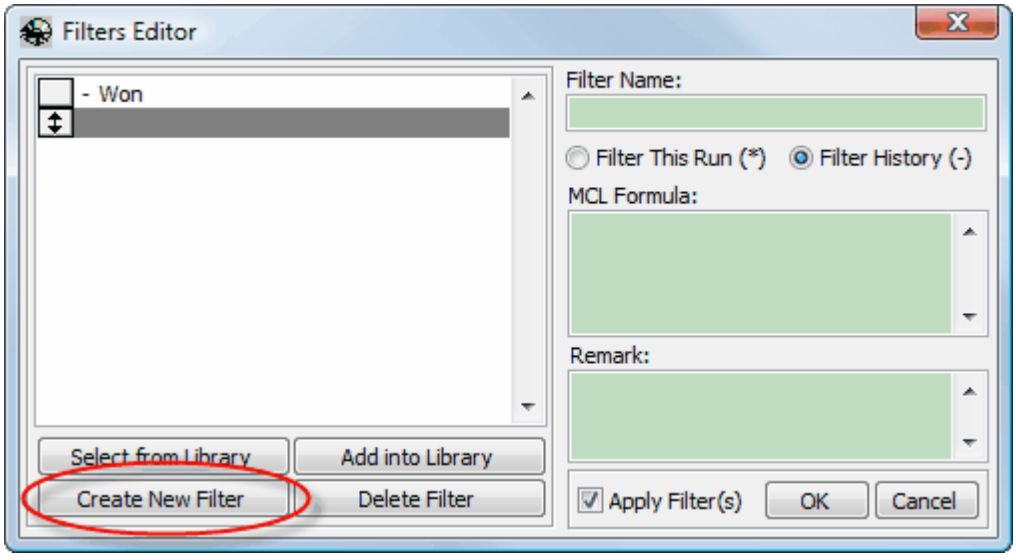
© Welkin Systems Limited

Library. Select a filter from the list and click on [Apply] as one of your filter items.




### Creating a Custom Filter

Click on the [Create New Filter] button on the Filter Editor. You may then create your own filter by entering an MCL formula.




### Adding Your Custom Filter into Library

Once you created a custom filter, you can click on the [Add into Library] button to add it into the Filter Library.

### Removing a Filter from the Filter Group

Click on the [Delete Filter] button to remove the highlighted filter.

### Choosing between [Filter This Run (*)] and [Filter History]

© Welkin Systems Limited

Choose [Filter This Run (*)] if you are going to filter starters of the current race

## 6.2 Type 2: History and Statistics Views

The [History/Stat] page of the Horse, Jockey and Stable forms are sharing the statistics engine of the "Comprehensive Statistics and Record Query" form. It can be used either for retrieving past records or performing statistics.

History and Statistics Views are different from Starter Views in many ways. This section will discuss the basic operations on the page.

See the Type 1: Comprehensive Statistics & Record Query section for more information of the statistics views.

**>>> Subtopics of this section <<<**

- Types of History and Statistics Views
- Argument of History/Stat View
- Setting Data Range in an Argument

### 6.2.1 Types of History and Statistics Views

**History and Statistics View Page of the Horse, Jockey and Stable Forms**

The following operations can be done on the [History/Stat] page of the Horse, Jockey and Stable Forms

1. Modify your own view. You may
   - add columns that you are interested in into a view
   - delete columns that you do not want
   - add any number of user notes columns
2. Set criteria to filter and query the records that you want
3. Carry out statistics
   - Apply statistics to selected columns
   - Do unlimited level of breakdown and grouping
   - Toggle between [Statistics] mode and [Runs] mode whenever necessary.
4. Export your results into Microsoft Excel
   - You can export your statistics results into a Microsoft Excel

**Views on the Comprehensive Statistics and Record Query Form**

The views on the "Comprehensive Statistics and Record Query Form" is basically identical to those on the [History/Stat] page of the Horse, Jockey and Stable forms. The only difference is the views on the [History/Stat] page will retrieve only records relevant to the form while the Comprehensive one will retrieve all past records of the period that you define.

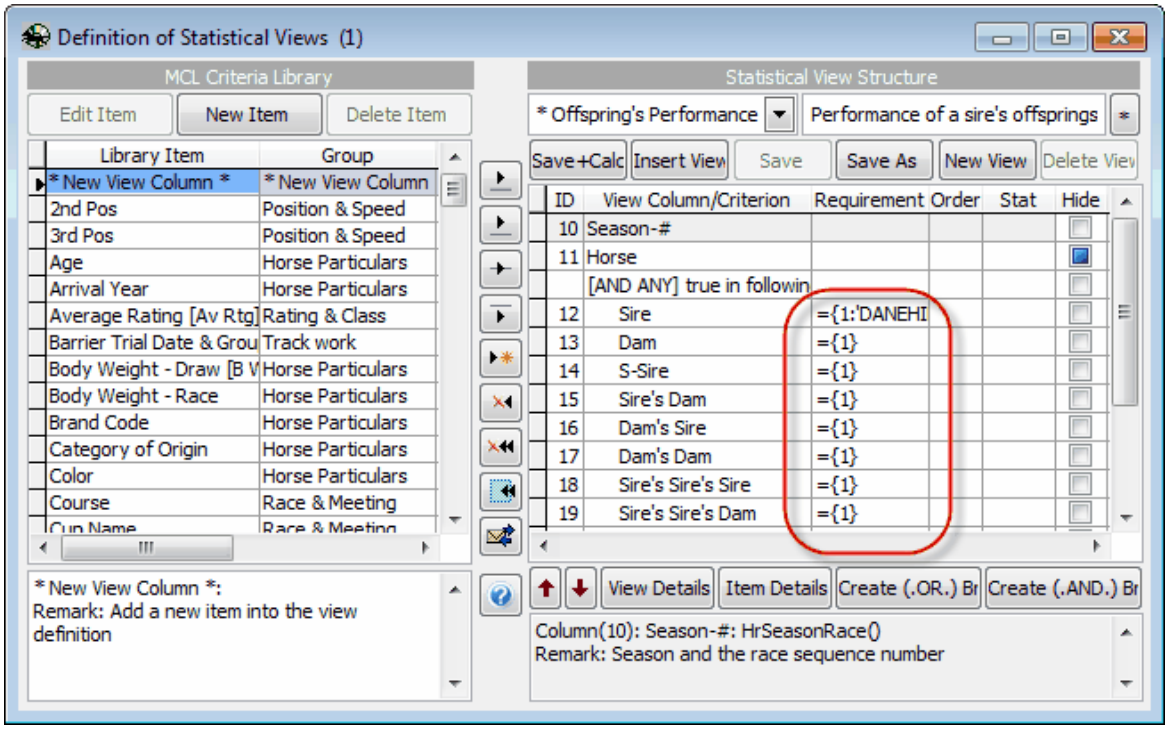
**Detailed Operations**

Since the elements and operations of the above 2 types of views are identical, please see the content of the page Comprehensive_Statistics_and_Record_Query

© Welkin Systems Limited

### 6.2.2 Argument of History/Stat View

Let's take "* Offsprings' Performance" as an example. The default value of MCL variable 1 is DANEHILL. We can change its value by passing another value to it. Open Using MCL Variables As View Arguments to see how to pass arguments to a history/statistics view.




**See Also**

Introduction to MCL Variables | Setting Values & Choosing MCL Variables

### 6.2.3 Setting Data Range in an Argument

Please see MCL Variables to Set Data Range.

## 6.3 Cache for View Data

**Data Caching**

It takes time for an MCL function to search and retrieve data from the database of RaceMate. Some statistics functions may need intensive calculations to return results. In order to speed up data display, RaceMate will put the data retrieved by functions into a cache folder. Next time when we open the same page, RaceMate will get the data directly from the cache without repeating the search and calculation processes. Although this dramatically speeds up data viewing, under some special circumstances, you may encounter data synchronization problems, for examples:

1. You have already retrieved data into a view (cached). However the data are updated afterwards. You will not be able to see the up-to-date data in your view.
2. You use your a conversion table in your calculation. You change the data in your conversion table after data are loaded into the cache. The new change will not be

© Welkin Systems Limited

shown.

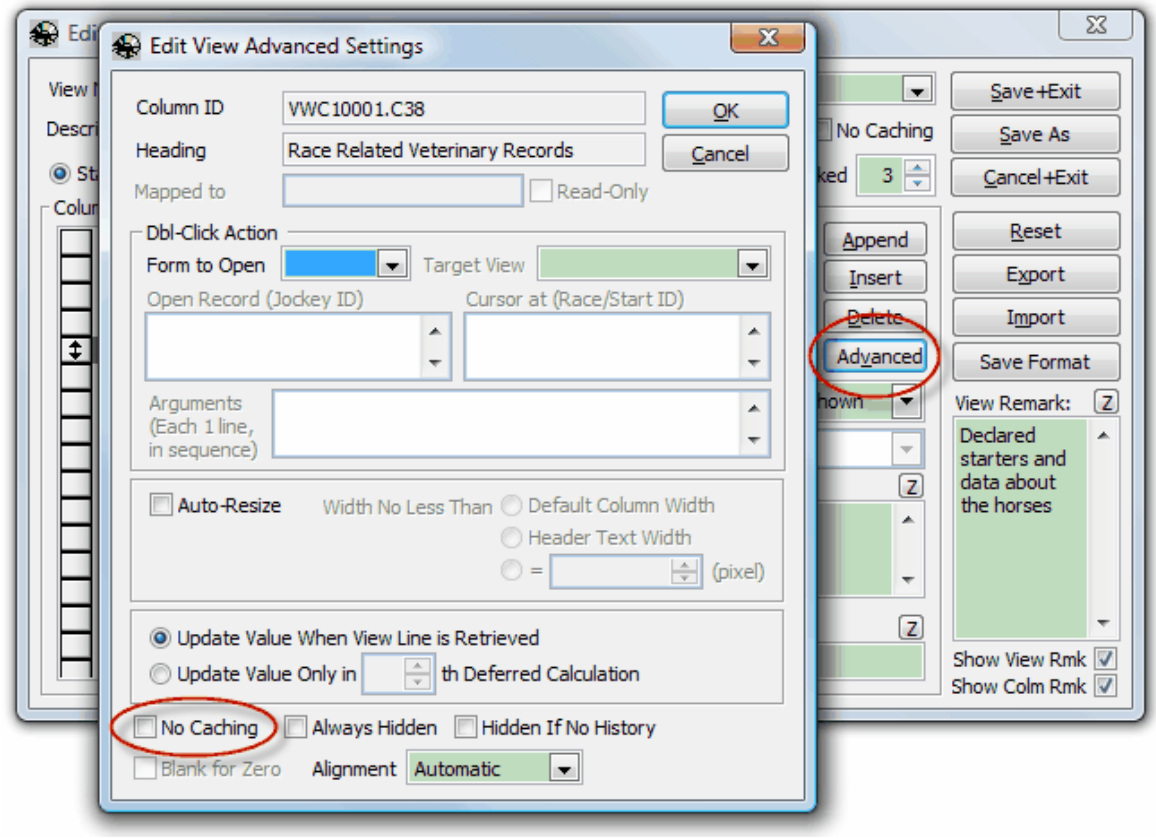
You may solve such data synchronization problem in the following ways.

### Manual Data Reload

Click on the [Recalc] button at the right hand side of your Starter Views page to reload data of the view of the current race.

### Set "No Caching" Attribute

You may set the "No Caching" attribute of column (advanced setting). RaceMate will not cache the data of a No Caching column and the column will be recalculated every time you open the race. However, never set every column as "No Caching" because it will affect the speed of the view.




### Clearing the Cache

Please see Clearing_Cache_Folder

## 6.4 Conversion Tables

We can use a conversion table to convert data from a particular value into another. A variety of MCL functions for retrieving the converted values are provided by RaceMate. Please see the section conversion_functions for details about the functions.

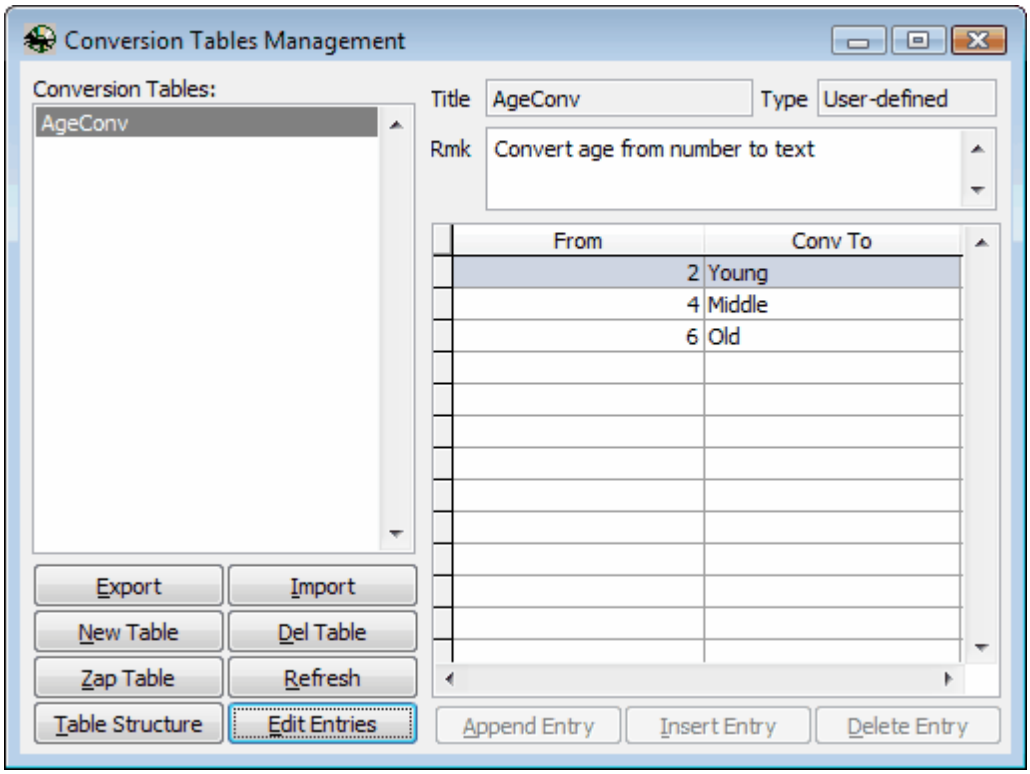
**>>> Subtopics of this section <<<**

© Welkin Systems Limited

- Managing Conversion Tables
- Using Conversion Tables in Views

## 6.4.1 Managing Conversion Tables

A conversion table provides rules for converting a value into another. Selecting [Conversion Table Management] from the [Analysis] menu of the RaceMate main menu. You will see the following form. The AgeConv table in the following figure converts 2 to "Young", 4 to "Middle" and 6 to "Old".



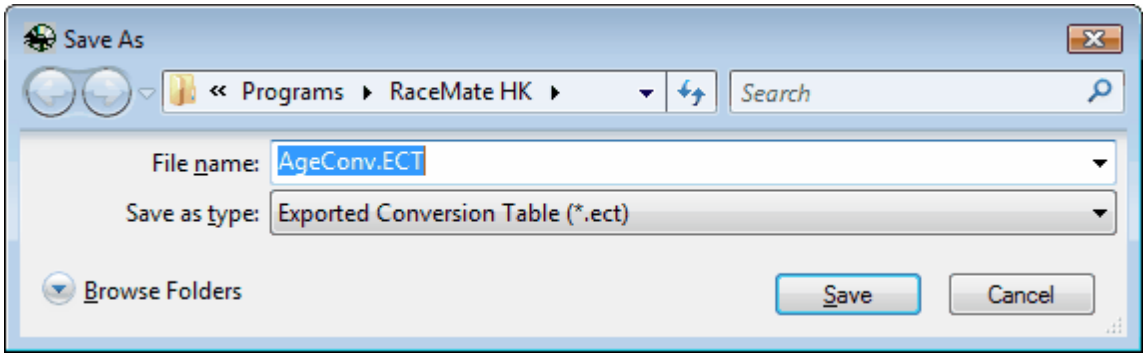

### Edit Table Content

Click on the [Edit Entries] button to edit the content of the conversion table. You may use the [Append Entry], [Insert Entry] and [Delete Entry] buttons to add and delete entries of the table.

### Export Conversion Table

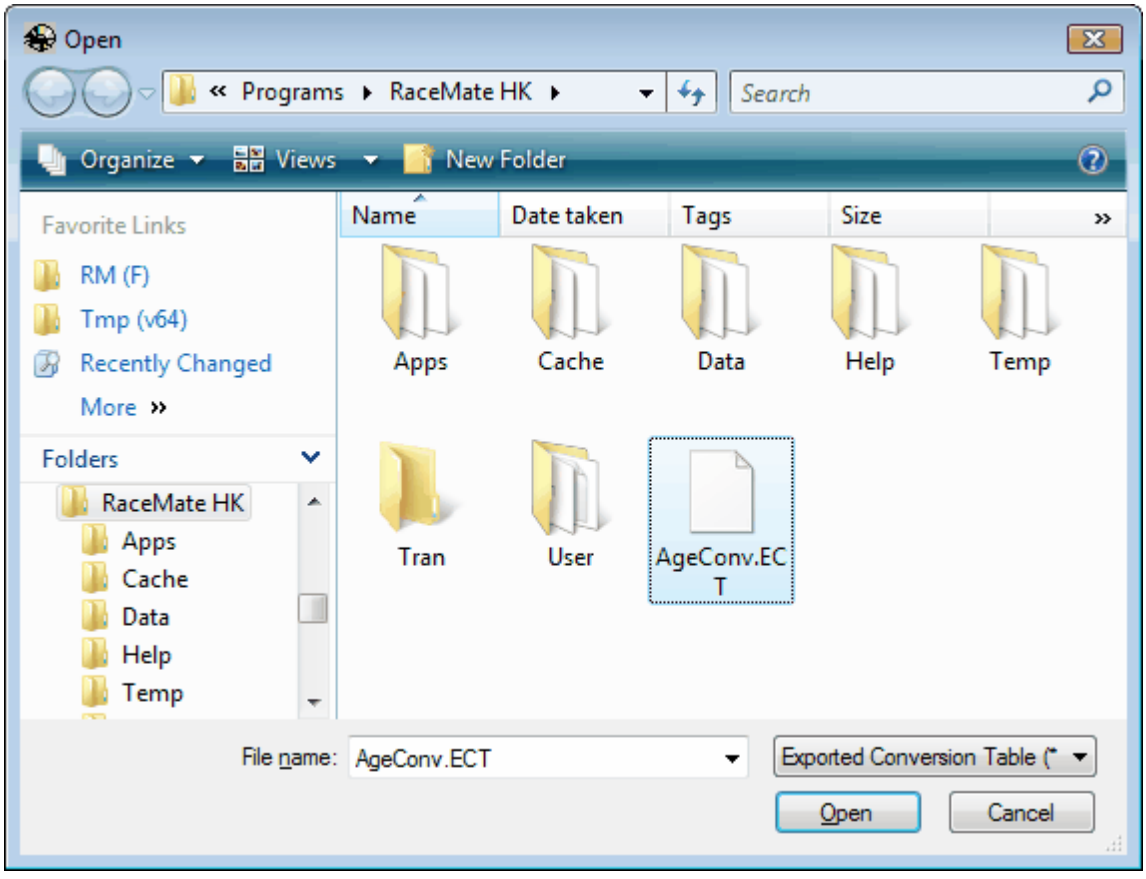
Click on the [Export] button. You will be prompted to assign a file name to your exported file.

© Welkin Systems Limited




**Import Conversion Table**

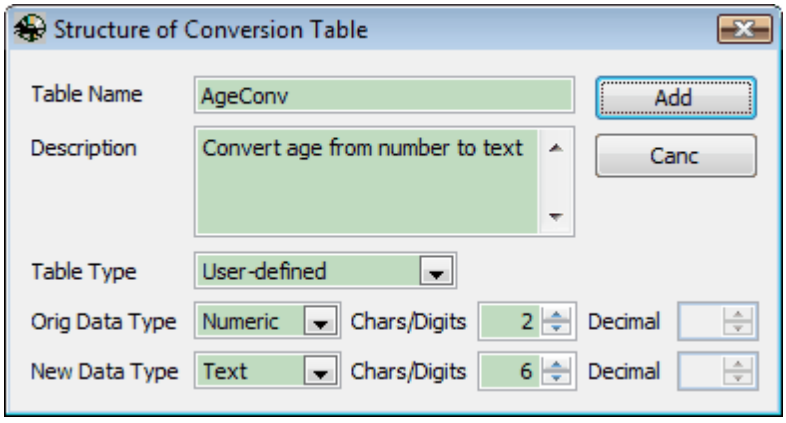
Click on the [Import] button. When you see the following dialogue, select the exported conversion table you want from the hard disk and click on [Open] to import the file.




**Create New Conversion Table**

Click on the [New Table] button to create a new conversion table.

© Welkin Systems Limited




**Delete Conversion Table**

Click on the [Del Table] button to delete the highlighted conversion table.

**Zap Conversion Table**

Click on the [Zap Table] button to clear the editing table.

**Modify Table Structure**

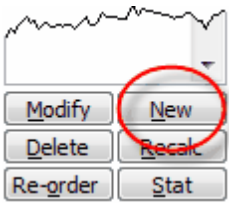
Click on the [Table Structure] button to show the "Structure of Conversion Table" form. On the form you can modify the structure of the table.

## 6.4.2 Using Conversion Tables in Views

Let us see how the conversion table created in the page Managing Conversion Tables is used in a view. Please note that this is only an example.

**Step 1**

Click on the [New] button below the view list to create a new view.




**Step 2**

Enter "Age Conversion" for as view name.

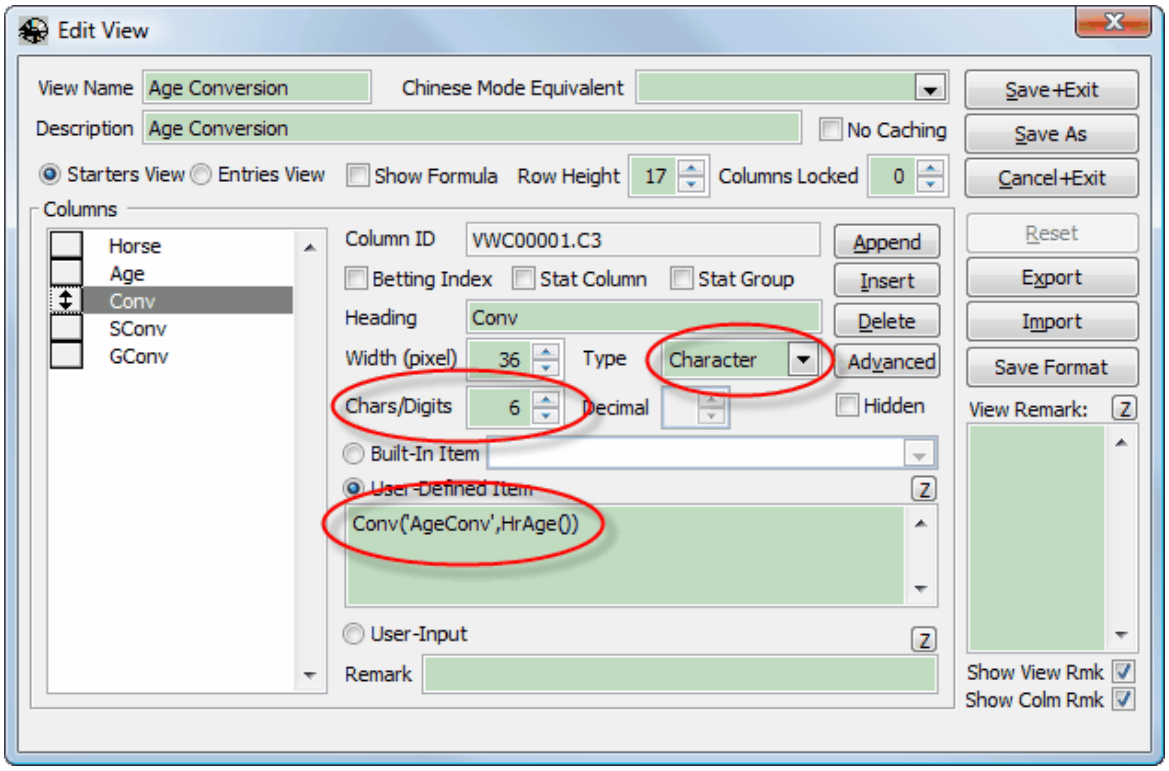
**Step 3**

Add the following columns for the view:

| Column Name | Type (width) | User-Define Formula | Remarks |
|---|---|---|---|
| Horse | | (built-in column) | It should be there because it exists when you create the view |

© Welkin Systems Limited

| | | | |
|---|---|---|---|
| Age | | (built-in column) | Click on the [Append] button to add this column |
| Conv | Character (6) | Conv('AgeConv',HrAge ()) | Click on the [Append] button to add this column |
| SConv | Character (6) | SConv('AgeConv', HrAge()) | Click on the [Append] button to add this column |
| GConv | Character (6) | GConv('AgeConv', HrAge()) | Click on the [Append] button to add this column |




**Result**

When you click on the [Save+Exit] button, you will see the following result.

| Horse | Age | Conv | SConv | GConv |
|---|---|---|---|---|
| Thunder Flash | 4 | .NULL. | Young | Middle |
| Tiger Ridge | 7 | Old | Old | Old |
| Good To Go | 5 | Middle | Middle | Middle |
| Brilliant One | 6 | .NULL. | Middle | Old |
| The World Of Love | 8 | .NULL. | Old | .NULL. |
| Wine Win | 5 | Middle | Middle | Middle |
| Dragon Eye | 3 | Young | Young | Young |
| Kimberley Mines | 9 | .NULL. | Old | .NULL. |
| Country Magic | 4 | .NULL. | Young | Middle |
| Classa For Ever | 7 | Old | Old | Old |
| Kingifly | 6 | .NULL. | Middle | Old |
| Galactic Tactic | 5 | Middle | Middle | Middle |

Here is why the results are returned:

| Function | Description |
|---|---|
| Conv('AgeConv',HrAge( )) | 1 to 1 conversion. Since 4, 6, 8 and 9 don't have any |

© Welkin Systems Limited

| | |
|---|---|
| | matching value, Null is returned. |
| SConv('AgeConv',HrAge( )) | Matching equals or smaller. 4 will match 3, 6 will match 5 and 8, 9 will match 7. |
| GConv('AgeConv',HrAge( )) | Matching equals or greater. 4 will match 5, 6 will match 7 and no match is found for 8 and 9. |

## 6.5 Linking Up Views

This chapter will discuss how to link up two different views so that in one view you can display and edit data of another view.

**>>> Subtopic of this section <<<**

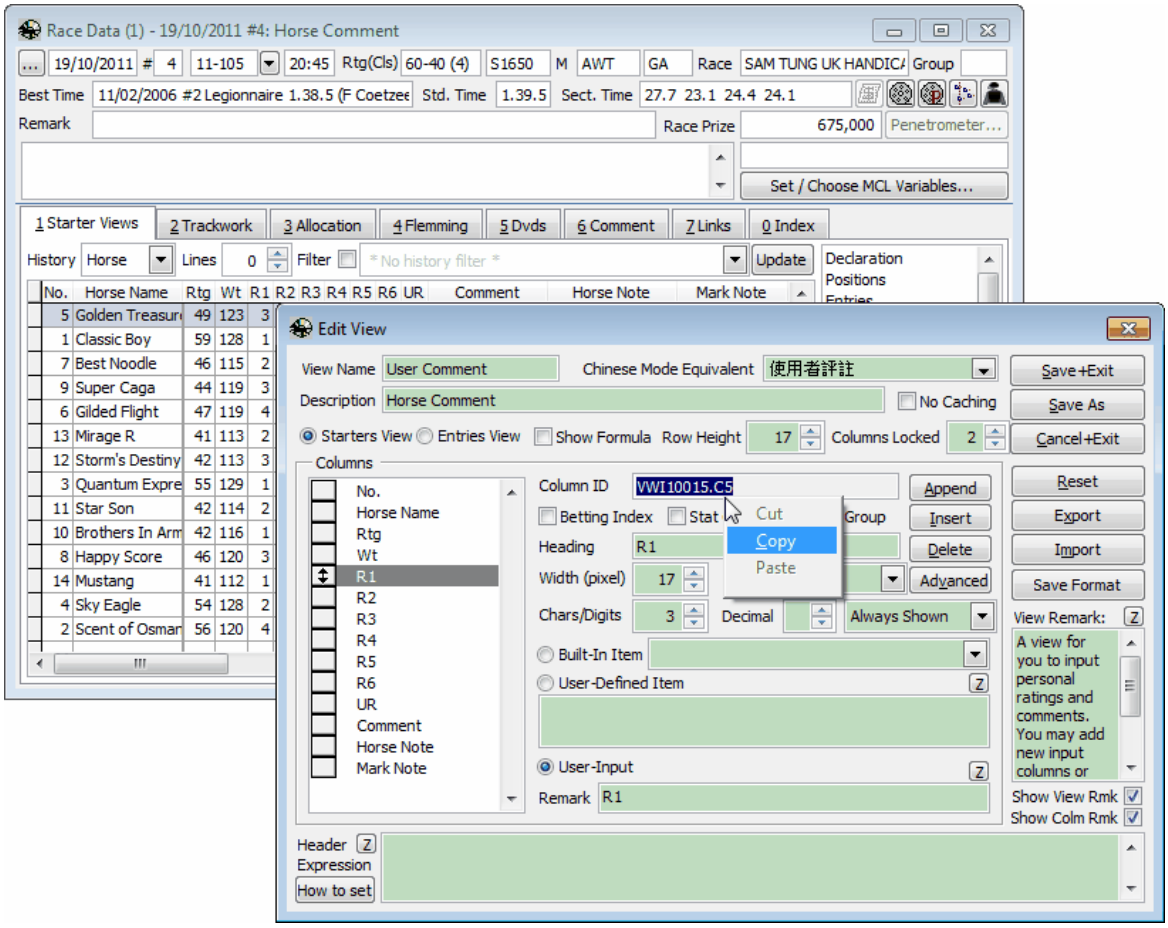
- Linked Columns

### 6.5.1 Linked Columns

You can display and edit data of another view in any view. The following example demonstrates how to duplicate the R1 column of the User Comments view in the Declaration view.

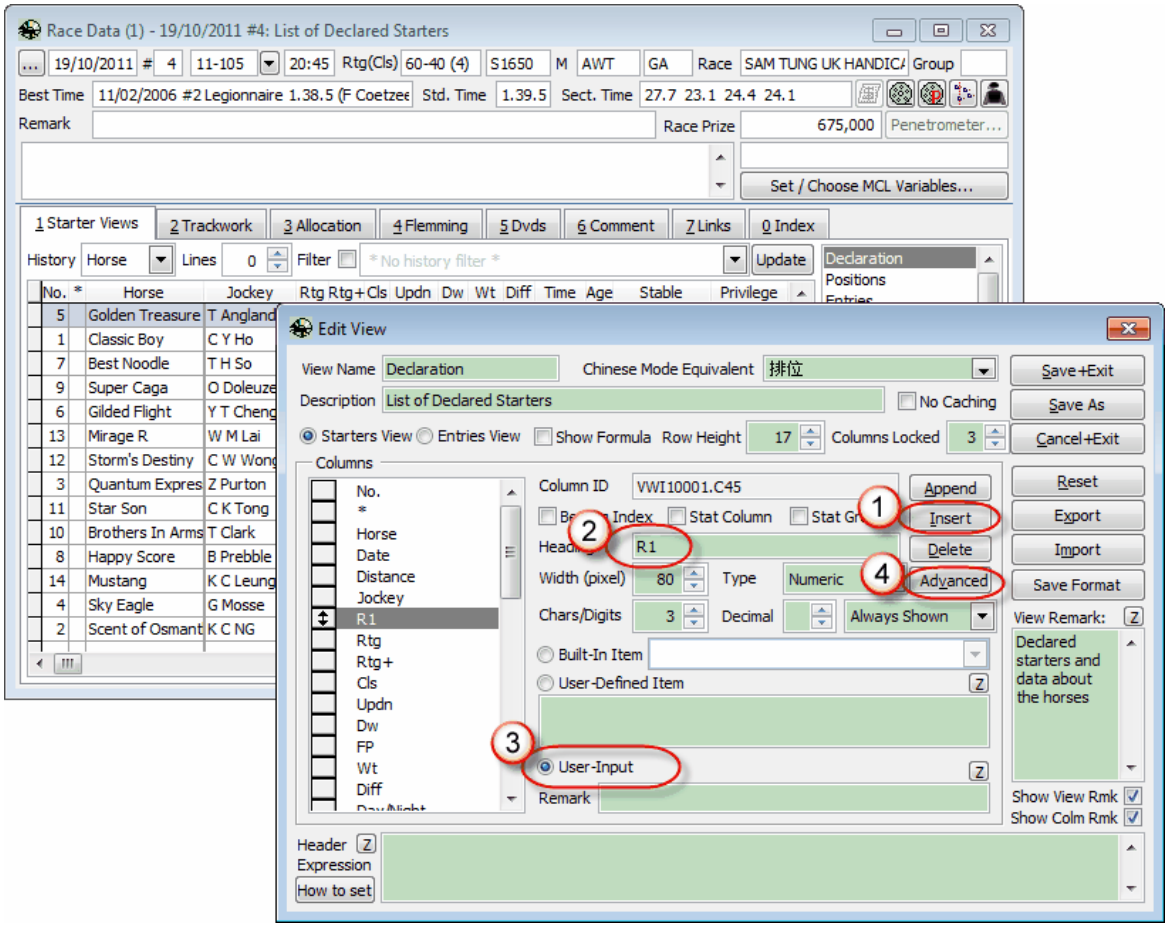
1. Open the User Comments view. Click on the [Modify] button at the right hand side to show the Edit View Form.

© Welkin Systems Limited

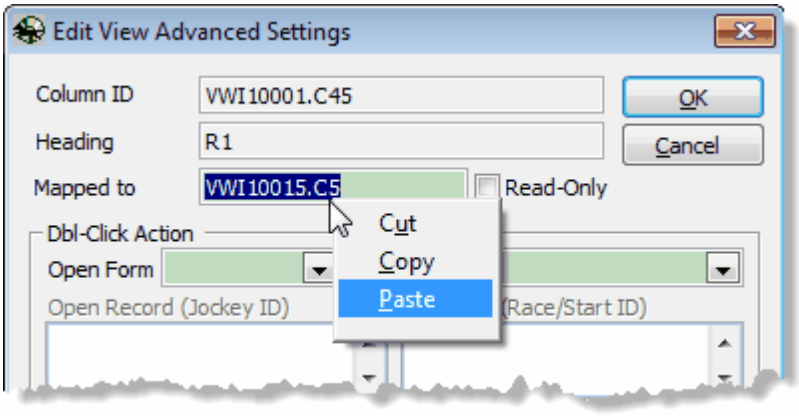



2. Highlight column R1. Copy it Column ID into Windows' clipboard.
3. Close the Edit View form and switch to the Declaration view.
4. Add a new column "R1" right after the "Rtg" column and set it as User-Defined Item.

© Welkin Systems Limited

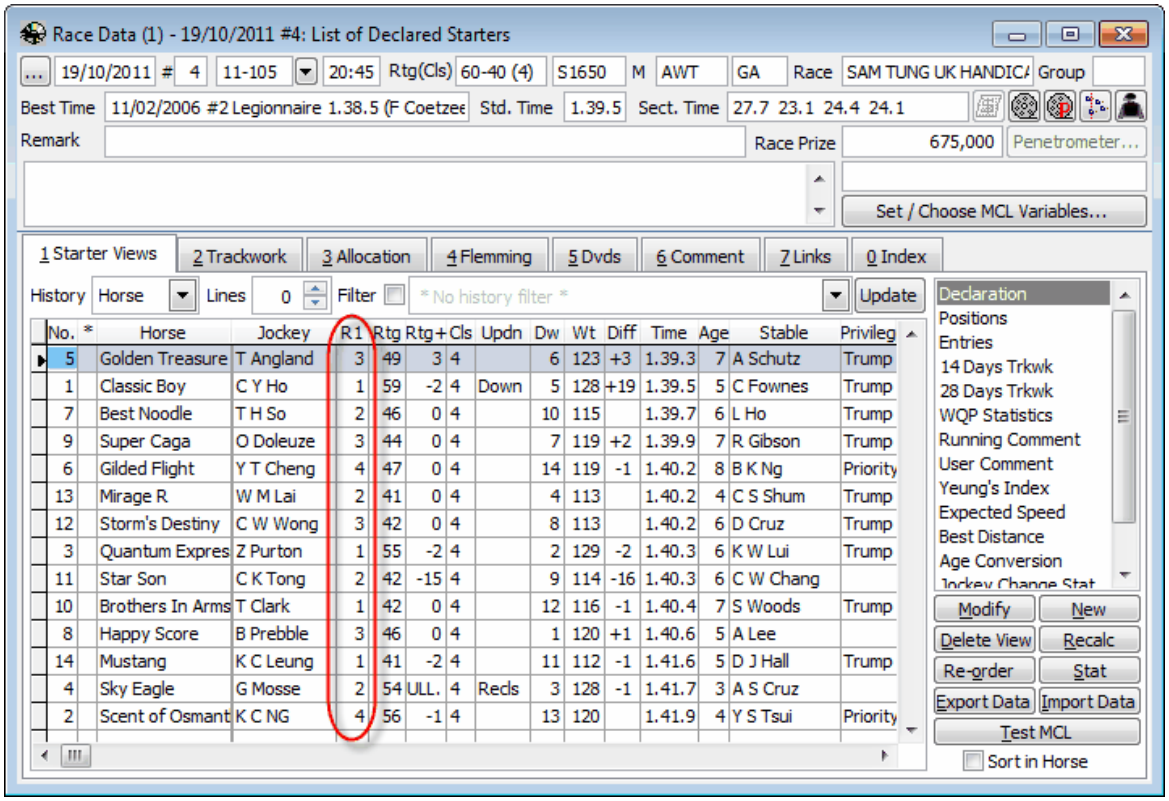



5. Click on the [Advanced] button to show the "Edit View Advanced Settings" form. Paste the Column ID that we copied from the User Comments view into the Linked Field text box.




6. Save the view definition and return to the Declaration view data grid. You will found the new R1 column inserted at the right hand side of the Rtg column.

© Welkin Systems Limited


Race Data (1) - 19/10/2011 #4: List of Declared Starters
19/10/2011 # 4 11-105 20:45 Rtg(Cls) 60-40 (4) S1650 M AWT GA Race SAM TUNG UK HANDICA Group
Best Time 11/02/2006 #2 Legionnaire 1.38.5 (F Coetzee Std. Time 1.39.5 Sect. Time 27.7 23.1 24.4 24.1
Remark
Race Prize 675,000 Penetrometer...
Set / Choose MCL Variables...
1 Starter Views 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Comment 7 Links 0 Index
History Horse Lines 0 Filter * No history filter * Update
No. * Horse Jockey R1 Rtg Rtg+ Cls Updn Dw Wt Diff Time Age Stable Privileg
5 Golden Treasure T Angland 3 49 3 4 6 123 +3 1.39.3 7 A Schutz Trump
1 Classic Boy C Y Ho 1 59 -2 4 Down 5 128 +19 1.39.5 5 C Fownes Trump
7 Best Noodle T H So 2 46 0 4 10 115 1.39.7 6 L Ho Trump
9 Super Caga O Doleuze 3 44 0 4 7 119 +2 1.39.9 7 R Gibson Trump
6 Gilded Flight Y T Cheng 4 47 0 4 14 119 -1 1.40.2 8 B K Ng Priority
13 Mirage R W M Lai 2 41 0 4 4 113 1.40.2 4 C S Shum Trump
12 Storm's Destiny C W Wong 3 42 0 4 8 113 1.40.2 6 D Cruz Trump
3 Quantum Expres Z Purton 1 55 -2 4 2 129 -2 1.40.3 6 K W Lui Trump
11 Star Son C K Tong 2 42 -15 4 9 114 -16 1.40.3 6 C W Chang
10 Brothers In Arms T Clark 1 42 0 4 12 116 -1 1.40.4 7 S Woods Trump
8 Happy Score B Prebble 3 46 0 4 1 120 +1 1.40.6 5 A Lee
14 Mustang K C Leung 1 41 -2 4 11 112 -1 1.41.6 5 D J Hall Trump
4 Sky Eagle G Mosse 2 54 ULL. 4 Recls 3 128 -1 1.41.7 3 A S Cruz
2 Scent of Osmant K C NG 4 56 -1 4 13 120 1.41.9 4 Y S Tsui Priority
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
User Comment
Yeung's Index
Expected Speed
Best Distance
Age Conversion
Modify New
Delete View Recalc
Re-order Stat
Export Data Import Data
Test MCL
Sort in Horse

Please note that no matter you edit data of R1 in either the Declaration view or the User Comments view, data in both views will be changed at the same time.

## 6.6 Exporting and Importing View Data

In RaceMate version 10.0 or later you can export and import data of views into Excel files. See the following topics for details.

- Exporting View Data
- Importing View Data From Excel Files

© Welkin Systems Limited

# Part VII

# 7 MCL Variables in MCL Functions

You can insert MCL variables into an MCL function. This allows us the rapidly replace the figures of a MCL function.

**>>> Subtopics of this section <<<**

- Introduction to MCL Variables
- Setting Values & Choosing MCL Variables
- Using MCL Variables As View Arguments
- MCL Variables to Set Data Range

## 7.1 Introduction to MCL Variables

You can embed variables in MCL functions. They will be replaced with either the default values or the values that you supply before the functions are evaluated.

### How to Define an MCL Variable and Its Default Value

An MCL variable is a number enclosed by a pair of quotes { and }. The number can be 1 to 9. For example {1} means variable number 1. You can put an optional default value by separating the variable number and the default value with a ":". For example {3:MyValue} defines a MCL variable {3} with a default value MyValue.

### MCL Variables of a View is Independent of MCL Variables of other Views

MCL variables of a view is independent of MCL variables of other views. You change the value of {2} in view X will not affect the value of {2} of view Y.

### Rules for Default Values of MCL Variable

It is a good practice to always set a default value for an MCL variable. The default value should be set in the first occurrence of the MCL variable.

### MCL Variables in Starter Views and History/Statistics Views

You can use MCL variables in both starter views and history/statistics views. However they are 2 independent sets of variable and have no relations with each other. Even if you have defined a value for {1} in a starter view, the value of it will not automatically passed to the history/statistics view. However you can pass its value to a history/statistics view via the rules discussed on the topic Using MCL Variables As View Arguments.

### Examples

```
HrActivity({3})
```

In the above function, MCL variable {3} is defined. Since there is no default value, you must later supply it with a value.

```
HrActivity({2:28})
```

In the above function, MCL variable {2} is defined with a default value 28. If no value is supplied to the variable, the function is equivalent to HrActivity(28).

© Welkin Systems Limited

```
JrFp(,"For HrHorse()='{1}'")
```

If you later set {1}=Brilliant Chapter, the above function will be equivalent to JrFp (,"For HrHorse()='Brilliant Chapter'")

**See Also**

Setting Values & Choosing MCL Variables, Using MCL Variables As View Arguments

## 7.2 Setting Values & Choosing MCL Variables

You can embed variables in MCL functions of a view. When the values of the MCL variables change, MCL functions will retrieve different data. See the help page Introduction to MCL Variables to learn more about MCL variables.

**Creating MCL Variables**

Set Introduction to MCL Variables for the steps.

**Editing MCL Variable Values**

If a MCL variable carries a default value, setting its value is optional. Click on the [Set / Choose MCL Variables...] button to show the MCL Variables editing form. A set of MCL variables is called an MCL variable group. You can edit the values and titles of existing variable groups. If you want to create a new group, please click on [Add New]. Click on [Delete] to remove a group. Order of groups can be rearranged by dragging the button on the left of the the group list.

**Show Title in Value Box**

Showing the group title on the Race Data form helps us to understand the data on the view. Checking the [Show Title in Value Box] will show the group title on the view data form while unchecking it will hide the title.

**Shared Value**

A MCL variable with the [Shared Value] box checked means the change in any group will also change the values of that variable in other groups.

**Selecting MCL Variable Groups**

Select a group title on the group list and click on the [Apply] button to apply the variable values to the current view. You may select more than one groups by checking the boxes on the group list. When more than one group is selected, RaceMate will follow the rules below in assigning values to MCL variables.

- If different values are assigned to the same variable in different groups, the lower group will take the precedence.
- If the [Use Blank] checkbox is not selected, the blank value will be ignored irregardless of the group position. If the [Use Blank] checkbox is selected, the blank value will be treated as a true value.

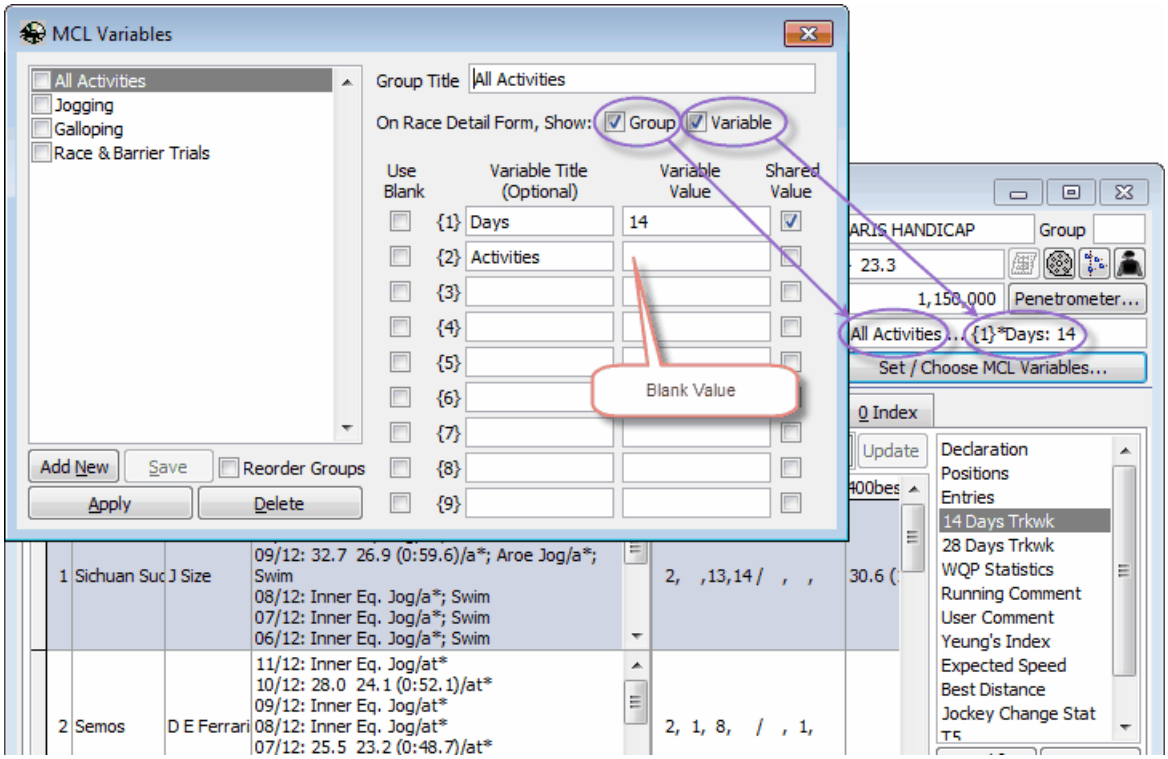
**Positions of MCL Variable Groups on Group List**

- You may check [Reorder Groups] to show the "reorder bars" on the group list. Drag

© Welkin Systems Limited

the bars to reorder the groups.

- If you apply more than one group of variables to a view and a variable, say {2}, have different values in different groups, RaceMate will use the value of the group at the lowest position of the group list.



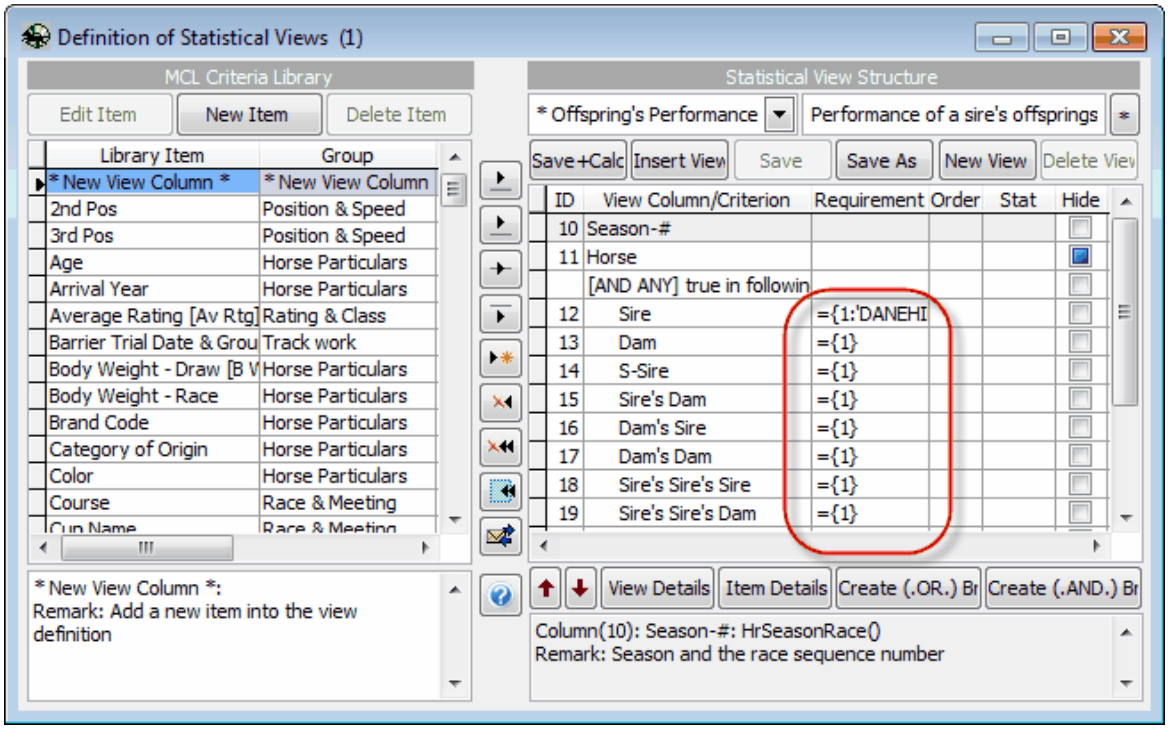

**See Also**

Introduction to MCL Variables | Using MCL Variables As View Arguments

## 7.3 Using MCL Variables As View Arguments

Open the structure of the "Offsprings' Performance" view. You will find an MCL variable {1}. The item {1:'DANEHILL'} means DANEHILL is the default value for the MCL variable {1}. You may change it to another default value. However the easiest way to supply this value is passing an argument value to the view.

© Welkin Systems Limited




You may set the following items on the form.

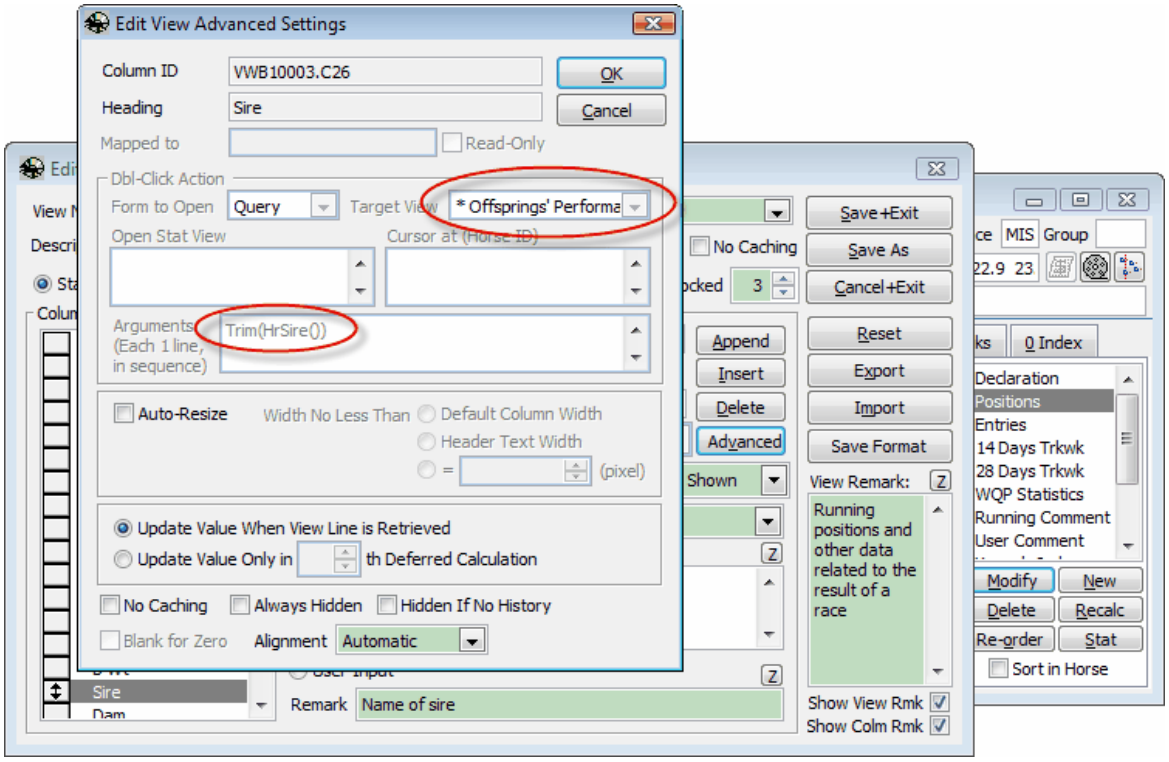
- Choose the a form on the [Open Form] drop down list
- Setting a view name on the [Target View] drop down list
- Arguments to be passed to MCL variables of the target view

If you edit the the property of the [Sire] column of of the [Position] view and then click on the [Advanced] button. You will see the following settings:

- Offsprings' Performance
- Trim(HrSire())

You can pass 9 arguments to a history/stat view to match 9 MCL variables (from {1} to {9}). Each argument must be put into a separate line in the argument editing box.

© Welkin Systems Limited



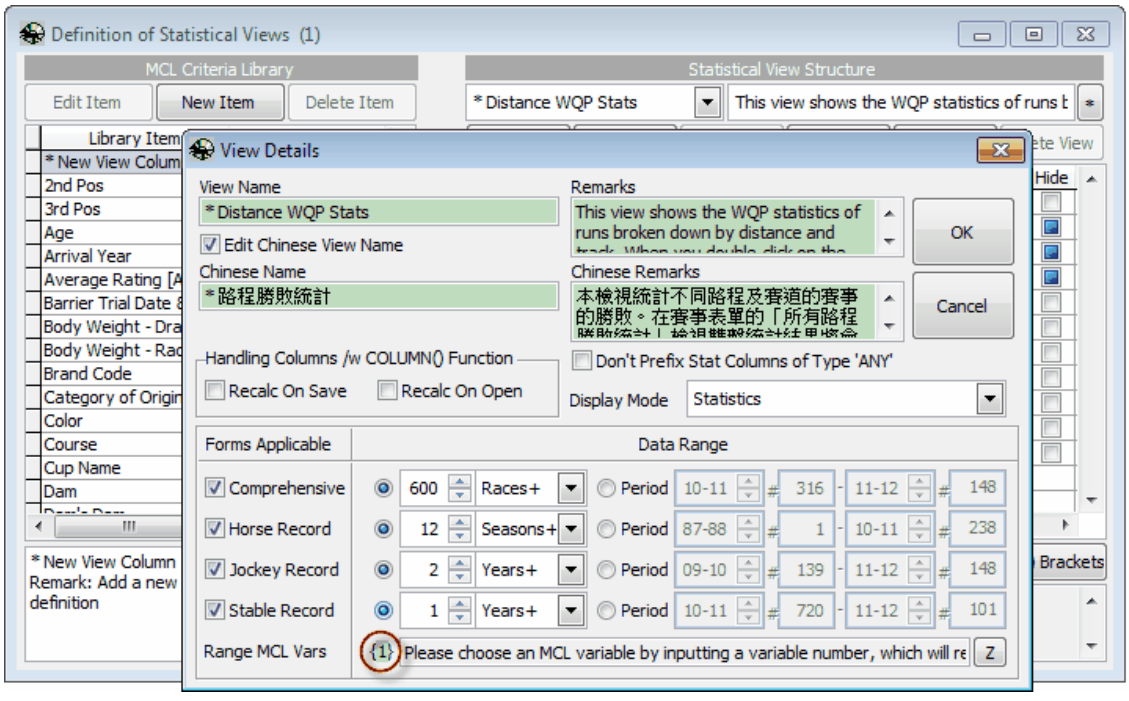

**See Also**

Introduction to MCL Variables | Setting Values & Choosing MCL Variables

## 7.4 MCL Variables to Set Data Range

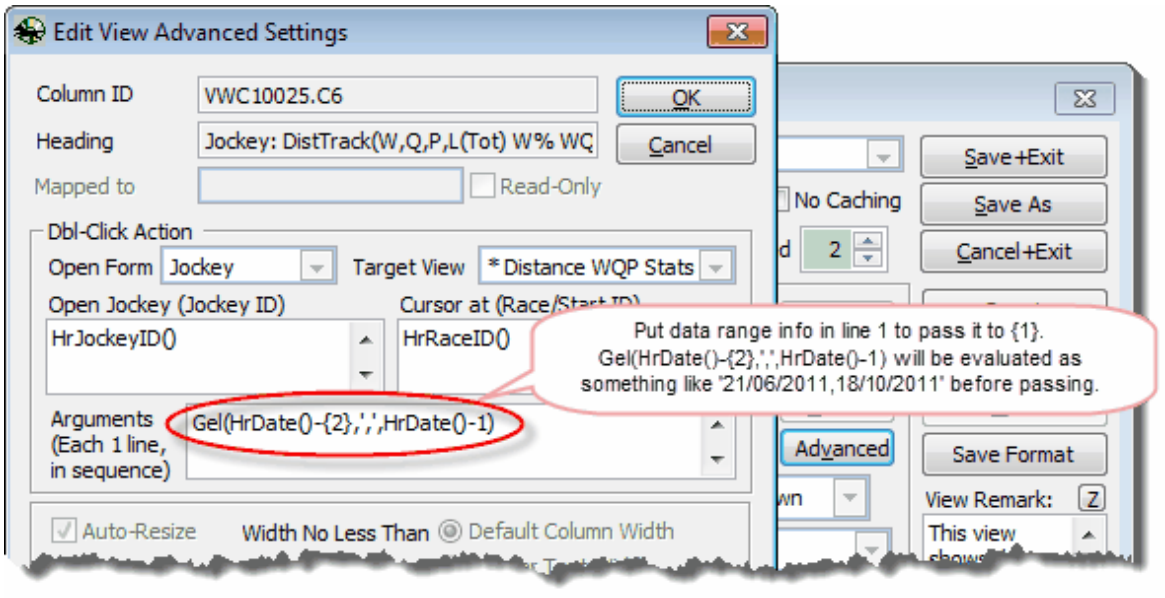
### Assigning an MCL Variable to Receive Data Range Information

On the bottom of the "View Details" form, you may assign an MCL variable to receive the data range information in the argument passed from a starter view.




© Welkin Systems Limited

When inputting the arguments into the "Arguments" box of the "Edit View Advance Settings" form, you should follow the one-to-one matching rule. That is if you have assigned MCL variable {1} to receive the information, the information should be put in line 1. If you have assigned {2}, put the information in line 2, and so on.




### Syntax for Data Range Argument

Use "3S" for 3 seasons and "3S+" for 3 seasons including future races. Replace S with Y, M, D, or R for year, month, day and race respectively. You may also pass "17/9/2010,19/12/2010" for a period between the dates, or pass "2010-1,2010-9999" for a period between race 1 and last race of season 2007-2008.

### See Also

Using MCL Variables As View Arguments | Introduction to MCL Variables | Setting Values & Choosing MCL Variables | Editing View Particulars

© Welkin Systems Limited

# Part VII

# 8 Built-in Views

This chapter explains the meanings of the views and their columns.

**>>> Subtopics of this section <<<**

- Built-in Starter Views

## 8.1 Built-in Starter Views

This chapter describes important built-in starter views.

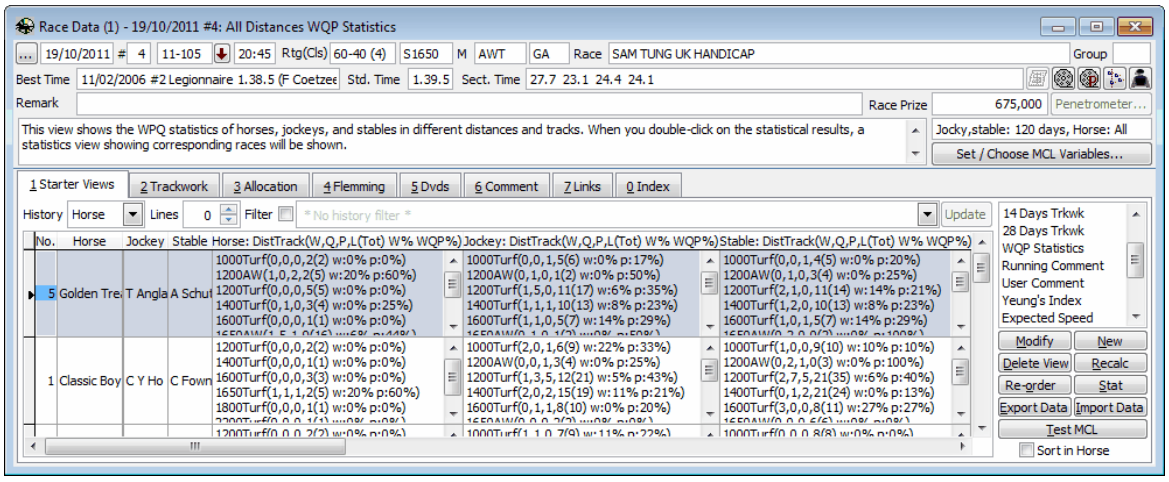
**>>> Subtopics of this section <<<**

- All Distances WQP Stats View
- Jockey Change Statistics

### 8.1.1 All Distances WQP Stats View

This view shows the W,Q,P counts of horses, jockeys and stables. You can double-click on a column to show the corresponding hist/stat view to verify the data on the column.

The screen capture below shows the figures for the 19/10/2011 meeting. We have set the statistical period to 120 days for jockeys and stables and all starts for horses.




**View Columns**

Columns: "No.", "Horse", "Jockey" and "Stable"

Shows horse no., horse name, jockey name and stable name.

Column: "Horse: DistTrack(W,Q,P,L(Tot) W% WQP%)"

This column shows the W,Q,P counts and the winning percentage of each horse. For example the line "1000Turf(0,0,0,2(2) w:0% p:0%)" on the top row of the column shows no winning, 2nd, and 3rd for "Golden Treasure". It has lost in 2 starts and the its total starts are 2. Both winning and 2nd percentage of it are 0.

Column: "Jockey: DistTrack(W,Q,P,L(Tot) W% WQP%)"

This column shows the W,Q,P counts and the winning percentage of each jockey

© Welkin Systems Limited

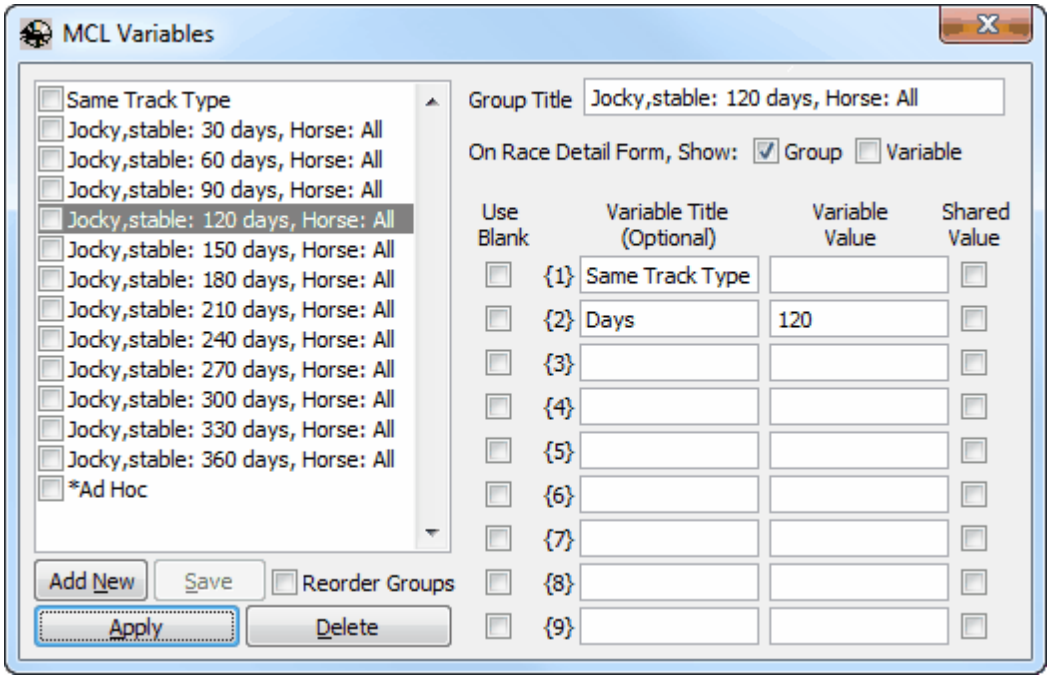
during the period defined by MCL variable {1}. For example the line "1000Turf(0,0,1,5 (6) w:0% p:17%)" on the top row of the column shows, during the latest 120 days, no winning and 2nd for "T Angland". He has one 3rd and lose in 5 races. His total races are 6. The winning percentage was 0 while the top-3 percentage of it are 17%.

Column: "Stable: DistTrack(W,Q,P,L(Tot) W% WQP%)"

This column shows the W,Q,P counts and the winning percentage of each stable during the period defined by MCL variable {1}. For example the line "1000Turf(0,0,1,4 (5) w:0% p:20%)" on the top row of the column shows, during the latest 120 days, no winning and 2nd for "A Schutz". He has one 3rd and lose in 4 races. His total races are 5. The winning percentage was 0 while the top-3 percentage of it are 20%.

### How to Change Statistical Period

You can change the statistical period by clicking on [Set/Choose MCL Variables...] on the right of the form. The "MCL Variables" form will be shown. Check the left-most boxes on the form to select an item or combine items. Learn more by opening the page Setting Values & Choosing MCL Variables.




### How Chaining is Set

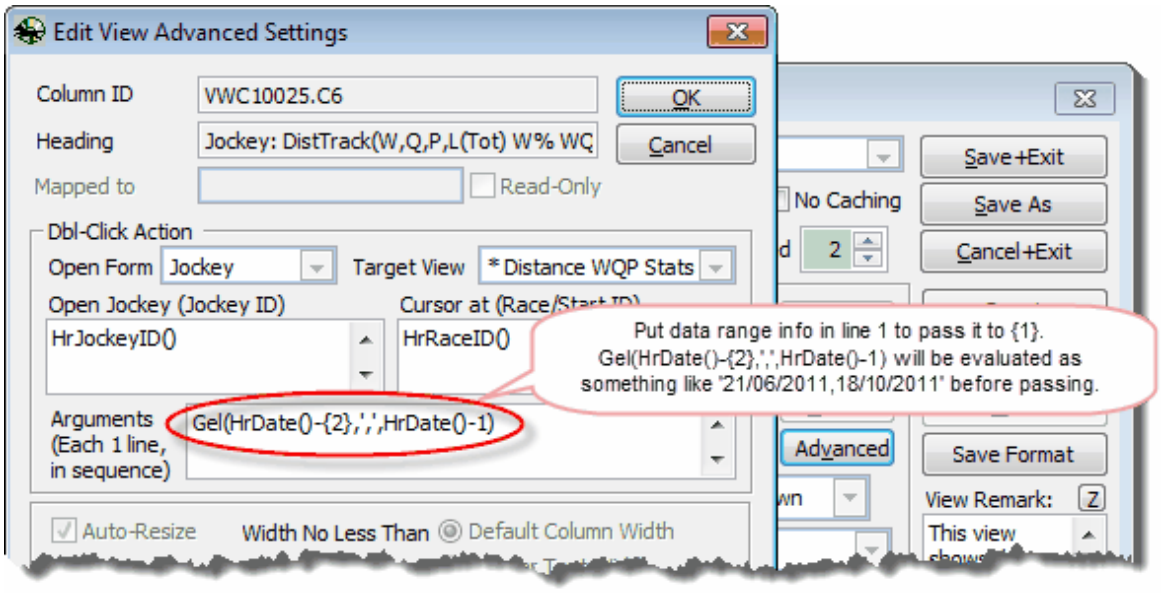
The statistics columns for Horse, Jockey and Stable have specific settings. When double-clicked, they will chain to the Horse, Jockey and Stable forms respectively and open the "* Distance WQP Stats" view. The data range will be set by argument 1. (See Using MCL Variables As View Arguments and MCL Variables to Set Data Range for details.)

Lets take "Jockey: DistTrack(W,Q,P,L(Tot) W% WQP%)" as an example. Highlight the column on the columns list at the left hand side of the "Edit View" form. The click the [Advanced] button at the right. You will see the following form.

| Field | Value | Remark |
|---|---|---|
| Open Form | Jockey | Open the "Jockey" form on double-clicking |
| Target View | * Distance WQP Stats | Jump to the "* Distance WQP |

© Welkin Systems Limited

| | | |
|---|---|---|
| | | Stats" view |
| Arguments | Gel(HrDate()-{2},',',HrDate()-1) | In this example {2} is 120. And HrDate() is 19/10/2011. Therefore<br>• HrDate()-{2} = 19/10/2011 - 120 = 21/06/2011<br>• HrDate()-1 = 19/10/2011 - 1 = 18/10/2011<br><br>The Gel() function combines the items (including the ',' inserted in the middle) into "21/06/2011,18/10/2011", which confirms the data range argument standard of RaceMate.<br><br>Since the argument if passed through argument 1, you must prepare {1} to receive it. |

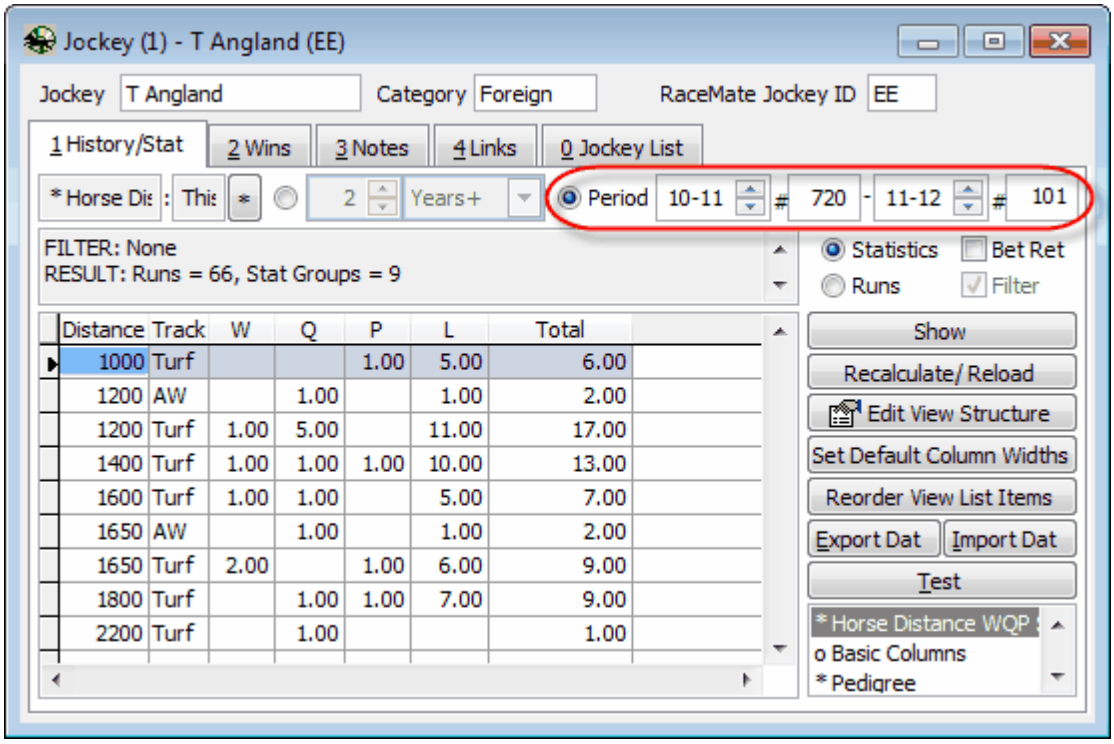



### History/Statistics Views after Double-Clicking

When one of the statistics columns is double-clicked, the form set on the "Edit View Advanced Settings" will be shown. The data range will be set according to the argument set there.

For example, when you double-click on the "Jockey: DistTrack(W,Q,P,L(Tot) W% WQP %)" column, the [* Distance WQP Stats] view of the Jockey form will be opened. The period is set to 21/06/2011 to 18/10/2011 which is equivalent to race 720 of season 10-11 to race 101 of season 11-12.

© Welkin Systems Limited

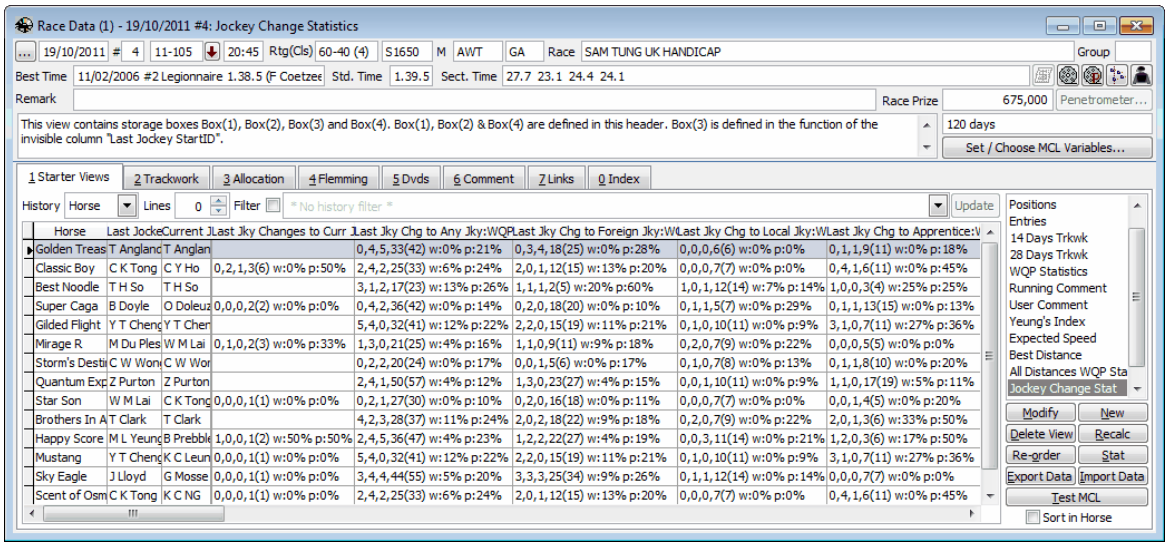



**See Also**

Using MCL Variables As View Arguments | MCL Variables to Set Data Range

## 8.1.2 Jockey Change Statistics

This view shows the WQPL statistics of jockey changes within a period. The period is defined by the MCL variable {1}. You may double-click on the statistical results to show all the runs covered by the statistics.


Race Data (1) - 19/10/2011 #4: Jockey Change Statistics
19/10/2011 # 4 11-105 20:45 Rtg(Cls) 60-40 (4) S1650 M AWT GA Race SAM TUNG UK HANDICAP Group
Best Time 11/02/2006 #2 Legionnaire 1.38.5 (F Coetzee Std. Time 1.39.5 Sect. Time 27.7 23.1 24.4 24.1
Remark
Race Prize 675,000 Penetrometer...
This view contains storage boxes Box(1), Box(2), Box(3) and Box(4). Box(1), Box(2) & Box(4) are defined in this header. Box(3) is defined in the function of the invisible column "Last Jockey StartID".
120 days
Set / Choose MCL Variables...
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Comment
7 Links
0 Index
History Horse Lines 0 Filter * No history filter * Update
Horse Last Jocke Current J Last Jky Changes to Curr J Last Jky Chg to Any Jky:WQP Last Jky Chg to Foreign Jky:W Last Jky Chg to Local Jky:W Last Jky Chg to Apprentice:W
Golden Treas T Angland T Anglan 0,4,5,33(42) w:0% p:21% 0,3,4,18(25) w:0% p:28% 0,0,0,6(6) w:0% p:0% 0,1,1,9(11) w:0% p:18%
Classic Boy C K Tong C Y Ho 0,2,1,3(6) w:0% p:50% 2,4,2,25(33) w:6% p:24% 2,0,1,12(15) w:13% p:20% 0,0,0,7(7) w:0% p:0% 0,4,1,6(11) w:0% p:45%
Best Noodle T H So T H So 3,1,2,17(23) w:13% p:26% 1,1,1,2(5) w:20% p:60% 1,0,1,12(14) w:7% p:14% 1,0,0,3(4) w:25% p:25%
Super Caga B Doyle O Doleuz 0,0,0,2(2) w:0% p:0% 0,4,2,36(42) w:0% p:14% 0,2,0,18(20) w:0% p:10% 0,1,1,5(7) w:0% p:29% 0,1,1,13(15) w:0% p:13%
Gilded Flight Y T Cheng Y T Cher 5,4,0,32(41) w:12% p:22% 2,2,0,15(19) w:11% p:21% 0,1,0,10(11) w:0% p:9% 3,1,0,7(11) w:27% p:36%
Mirage R M Du Ples W M Lai 0,1,0,2(3) w:0% p:33% 1,3,0,21(25) w:4% p:16% 1,1,0,9(11) w:9% p:18% 0,2,0,7(9) w:0% p:22% 0,0,0,5(5) w:0% p:0%
Storm's Desti C W Won C W Wor 0,2,2,20(24) w:0% p:17% 0,0,1,5(6) w:0% p:17% 0,1,0,7(8) w:0% p:13% 0,1,1,8(10) w:0% p:20%
Quantum Exp Z Purton Z Purton 2,4,1,50(57) w:4% p:12% 1,3,0,23(27) w:4% p:15% 0,0,1,10(11) w:0% p:9% 1,1,0,17(19) w:5% p:11%
Star Son W M Lai C K Tong 0,0,0,1(1) w:0% p:0% 0,2,1,27(30) w:0% p:10% 0,2,0,16(18) w:0% p:11% 0,0,0,7(7) w:0% p:0% 0,0,1,4(5) w:0% p:20%
Brothers In A T Clark T Clark 4,2,3,28(37) w:11% p:24% 2,0,2,18(22) w:9% p:18% 0,2,0,7(9) w:0% p:22% 2,0,1,3(6) w:33% p:50%
Happy Score M L Yeung B Prebble 1,0,0,1(2) w:50% p:50% 2,4,5,36(47) w:4% p:23% 1,2,2,22(27) w:4% p:19% 0,0,3,11(14) w:0% p:21% 1,2,0,3(6) w:17% p:50%
Mustang Y T Cheng K C Leun 0,0,0,1(1) w:0% p:0% 5,4,0,32(41) w:12% p:22% 2,2,0,15(19) w:11% p:21% 0,1,0,10(11) w:0% p:9% 3,1,0,7(11) w:27% p:36%
Sky Eagle J Lloyd G Mosse 0,0,0,1(1) w:0% p:0% 3,4,4,44(55) w:5% p:20% 3,3,3,25(34) w:9% p:26% 0,1,1,12(14) w:0% p:14% 0,0,0,7(7) w:0% p:0%
Scent of Osm C K Tong K C NG 0,0,0,1(1) w:0% p:0% 2,4,2,25(33) w:6% p:24% 2,0,1,12(15) w:13% p:20% 0,0,0,7(7) w:0% p:0% 0,4,1,6(11) w:0% p:45%
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comment
User Comment
Yeung's Index
Expected Speed
Best Distance
All Distances WQP Sta
Jockey Change Stat
Modify
New
Delete View
Recalc
Re-order
Stat
Export Data
Import Data
Test MCL
Sort in Horse

**View Columns**

Horse

Horse name

Last Jockey

Name of jockey who rides the same horse in the horse's last start

Current Jockey

The horse's current jockey

Last Jky Changes to Curr Jky:WQPL(T)

Shows the WQPL counts after a jockey change from last jockey to the current jockey within period set by MCL variable {1}.

In the screen capture above, the 2nd line of the "Last Jky Changes to Curr Jky:WQPL (T)" column shows that in latest 120 days, jockey change from C K Tong to C Y Ho occurs 5 times, among which the WQPL counts are 0,2,1,3 respectively. The winning rate is 0% while the top-3 rate is 50%.

Last Jky Chg to Any Jky:WQPL(T)

Shows the WQPL counts after a jockey change from last jockey to a jockey of any type within period set by MCL variable {1}.

In the screen capture above, the 1st line of the "Last Jky Chg to Any Jky:WQPL(T)" column shows that in latest 120 days, jockey changes from T Angland to other jockeys (any type) occur 35 times, among which the WQPL counts are 0,4,5,30 respectively. The winning rate is 0% while the top-3 rate is 23%.

Last Jky Chg to Foreign Jky:WQPL(T)

Shows the WQPL counts after a jockey change from last jockey to a foreign jockey within period set by MCL variable {1}.

In the screen capture above, the 1st line of the "Last Jky Chg to Foreign Jky:WQPL (T)" column shows that in latest 120 days, jockey changes from T Angland to other foreign jockeys occur 20 times, among which the WQPL counts are 0,2,4,14 respectively. The winning rate is 0% while the top-3 rate is 30%.

Last Jky Chg to Local Jky:WQPL(T)

Shows the WQPL counts after a jockey change from last jockey to a local jockey within period set by MCL variable {1}.

In the screen capture above, the 1st line of the "Last Jky Chg to Local Jky:WQPL(T)" column shows that in latest 120 days, jockey changes from T Angland to other local jockeys occur 5 times, among which the WQPL counts are 0,0,0,5 respectively. The winning rate is 0% while the top-3 rate is also 0%.

Last Jky Chg to Apprentice:WQPL(T)

Shows the WQPL counts after a jockey change from last jockey to an apprentice within period set by MCL variable {1}.

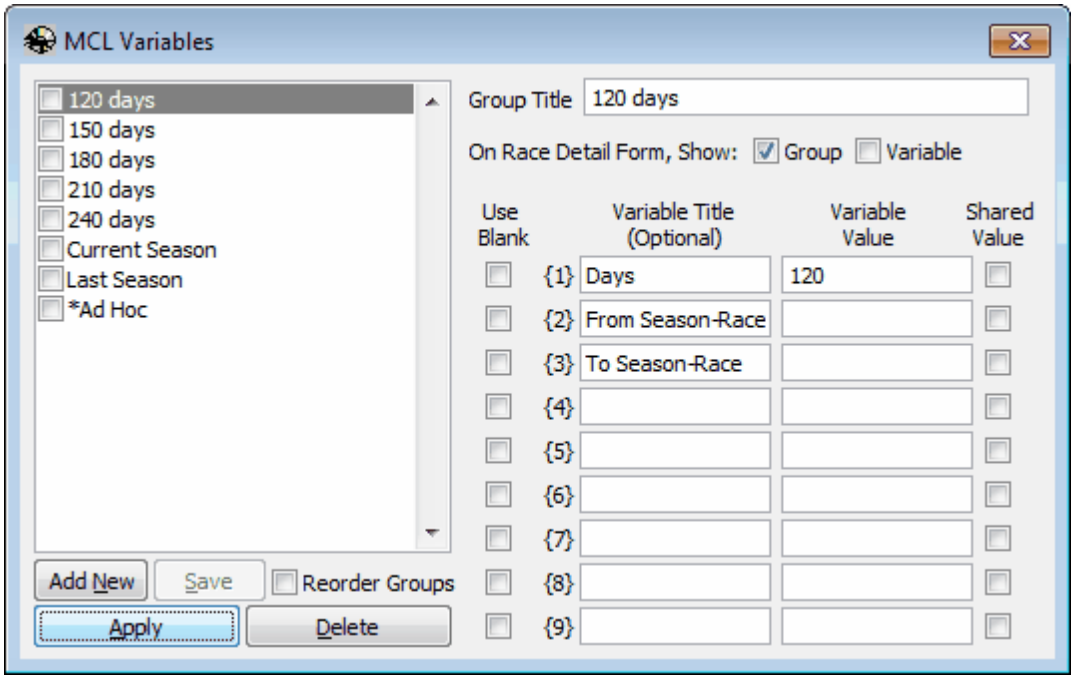
In the screen capture above, the 1st line of the "Last Jky Chg to Apprentice:WQPL(T)" column shows that in latest 120 days, jockey changes from T Angland to an apprentice occur 10 times, among which the WQPL counts are 0,1,1,8 respectively. The winning rate is 0% while the top-3 rate is 20%.

**How to Change Statistical Period**

You can change the statistical period by clicking on [Set/Choose MCL Variables...] on

© Welkin Systems Limited

the right of the form. The "MCL Variables" form will be shown. Check the left-most boxes on the form to select an item or combine items. Learn more by opening the page Setting Values & Choosing MCL Variables.




### How Chaining is Set

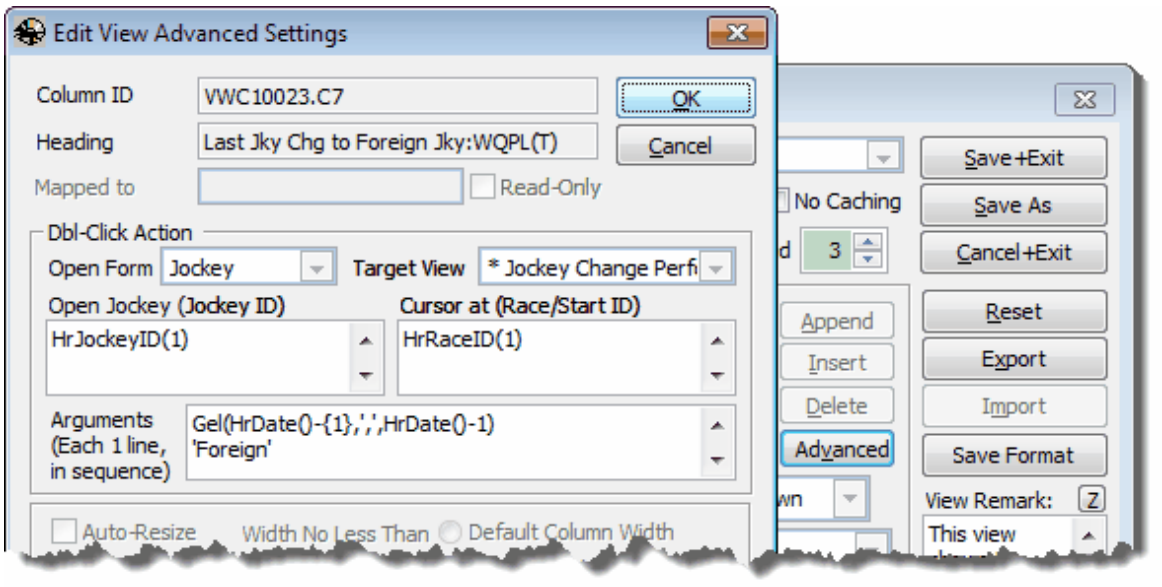
The statistics columns for Jockey Changes have specific settings. When double-clicked, they will chain to the Jockey forms showing the "* Jockey Change Performance" view. The data range will be set by argument 1. (See Using MCL Variables As View Arguments about how to pass data range.). Changing to which jockey type will be set by argument 2. Name of next jockey is set by argument 3. (Please note that column "Last Jky Chg to Any Jky:WQPL(T)" has argument 3 but no argument 2 while other statistics columns has argument 2 but no argument 3.)

Lets take "Last Jky Chg to Foreign Jky:WQPL(T)" as an example. Highlight the column on the columns list at the left hand side of the "Edit View" form. The click the [Advanced] button at the right. You will see the following form.

| Field | Value | Remark |
|---|---|---|
| Open Form | Jockey | Open the "Jockey" form on double-clicking |
| Target View | * Jockey Change Performance | Jump to the "* Jockey Change Performance" view |
| Arguments 1 | Gel(HrDate()-{1},',',HrDate()-1) | In this example {1} is 120. And HrDate() is 19/10/2011. Therefore<br>• HrDate()-{1} = 19/10/2011 - 120 = 21/06/2011<br>• HrDate()-1 = 19/10/2011 - 1 = 18/10/2011<br><br>The Gel() function combines the items (including the ',' inserted in the middle) into "21/06/2011,18/10/2011", |

© Welkin Systems Limited

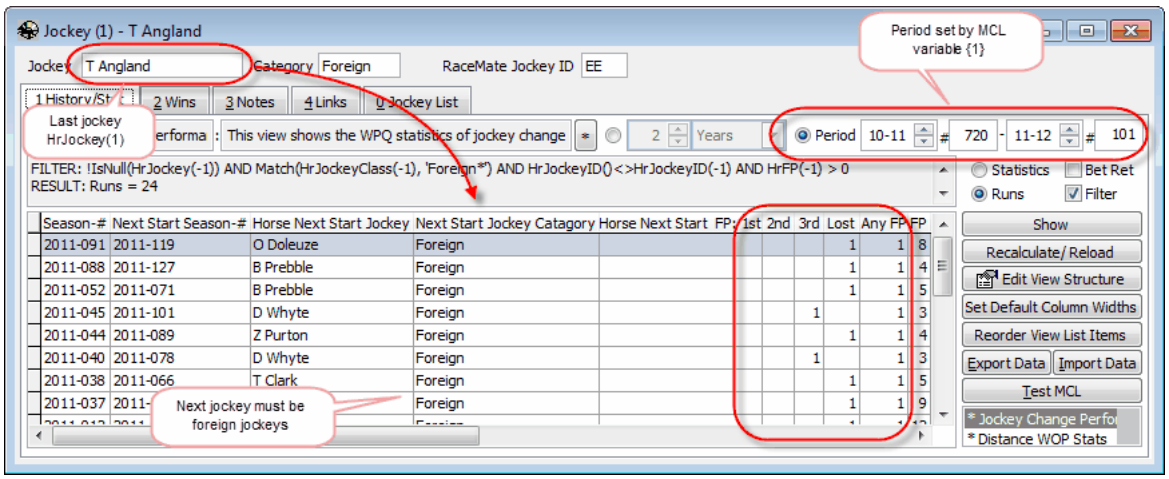
| | | |
|---|---|---|
| | | which confirms the data range argument standard of RaceMate.<br><br>Since the argument if passed through argument 1, you must prepare {1} to receive it. |
| Argments 2 | 'Foreign' | Set filter jockey type of next jockey to 'Foreign' |




### How to Show Races of the Statistics

Double-click on any of the statistical column open the "Jockey" form to show the runs of the corresponding statistics in a hist/stat view. Jockey type, name and period you that choose will be passed to the target view as arguments.

Let us take double-clicking on the first row of the column "Last Jky Chg to Foreign Jky: WQPL(T)" as an example. The value shown on the row is "0,3,4,18(25) w:0% p:28%". You can verify these figures on the "*Jockey Change Performance" view of the Jockey form.


Jockey (1) - T Angland
Period set by MCL variable {1}
Jockey
T Angland
Category
Foreign
RaceMate Jockey ID
EE
2 Wins
3 Notes
4 Links
Jockey List
Last jockey HrJockey(1)
This view shows the WPQ statistics of jockey change
Years
Period
10-11
720
11-12
101
FILTER: !IsNull(HrJockey(-1)) AND Match(HrJockeyClass(-1), 'Foreign*') AND HrJockeyID()<>HrJockeyID(-1) AND HrFP(-1) > 0
RESULT: Runs = 24
Statistics
Bet Ret
Runs
Filter
Season-# Next Start Season-# Horse Next Start Jockey Next Start Jockey Catagory Horse Next Start FP: 1st 2nd 3rd Lost Any FP FP
2011-091 2011-119 O Doleuze Foreign 1 1 8
2011-088 2011-127 B Prebble Foreign 1 1 4
2011-052 2011-071 B Prebble Foreign 1 1 5
2011-045 2011-101 D Whyte Foreign 1 1 3
2011-044 2011-089 Z Purton Foreign 1 1 4
2011-040 2011-078 D Whyte Foreign 1 1 3
2011-038 2011-066 T Clark Foreign 1 1 5
2011-037 2011- Foreign 1 1 9
Next jockey must be foreign jockeys
Show
Recalculate/ Reload
Edit View Structure
Set Default Column Widths
Reorder View List Items
Export Data
Import Data
Test MCL
* Jockey Change Perfo
* Distance WOP Stats

© Welkin Systems Limited

**See Also**

Using MCL Variables As View Arguments | MCL Variables to Set Data Range

© Welkin Systems Limited

# Part

# 9 Excel Spreadsheet Features

A RaceMate view is a spreadsheet similar to Microsoft Excel.

- You may do calculations among columns.
- On the sheet you may retrieve past or post records of horses, jockeys and stables.
- You may put statistics on cells.
- You may import data from external files.

**>>> Subtopics of this section <<<**

- How to Create Complicated Formulae
- MCL Debugging

## 9.1 How to Create Complicated Formulae

Once there is a mistake in your formula, you will see .Null. values in your column cells. If your formula are not a simple one, you may not be able to easily find out the cause of the problem. In general, when creating a complicated formula, we will follow the following rules.

- Rather than building the complete formula in one time, you should write a simple part of it first.
- If the simple part returns correct values, you will then add more to the formula and do the test again.
- This add and test process should be repeat and repeat until a complete formula is formed.

## 9.2 MCL Debugging

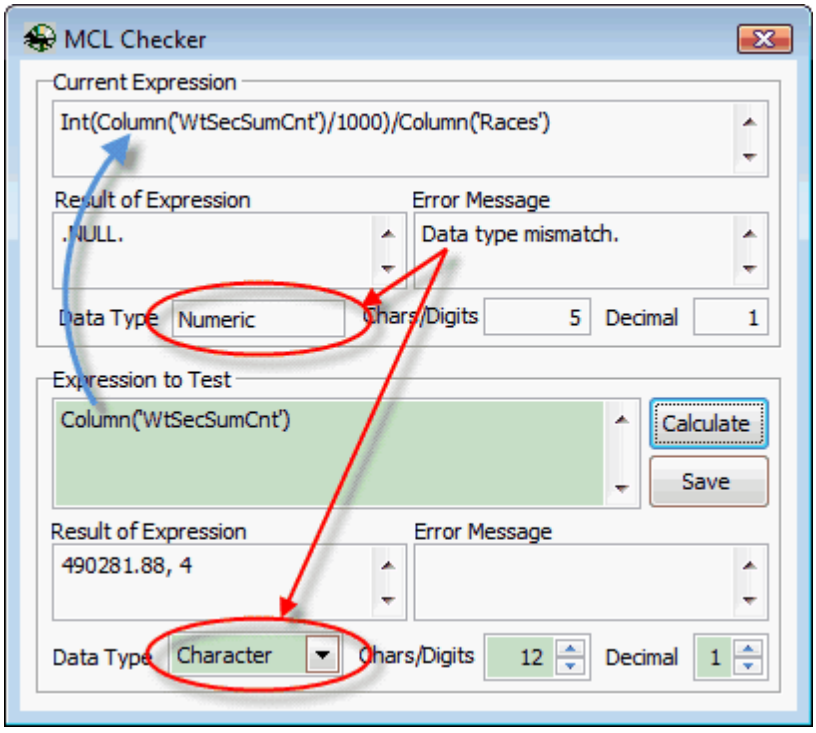
**Structure of MCL Formula Checker**

Clicking on the [Test MCL] button on the left bottom of the view page will show the 'MCL Checker' form. Before opening the form, you must click on a cell of the column the formula of which is going to be tested.

The MCL Checker is consisted of the following 2 parts.

1. Upper half - Current Expression
   The upper half of the form shows the current MCL expression. It also shows the expression result, data type, number of characters and error message, if any, etc.
2. Lower half - Expression to Test
   The lower half of the form is for testing a revised (or another) MCL expression. While the form shows, it copies the current expression into the "Expression to Test" box. You may delete some characters from the expression to make the expression simpler and then click on the [Calculate] button to check its result and data type.

© Welkin Systems Limited




**What Make the MCL Checker a Good Testing Tool?**

You will find the following strong points of the tool.

1. It is a very efficient tool because it calculates only a cell rather than the whole view.
2. It will tell you the data type of the expression to be checked.
3. It will suggest you the number of characters or digits (However you can't 100% rely on the suggestion. Sometimes you need to change the suggested values.)

**How to Test a Formula?**

You should choose a simple part of a complicated formula to test and confirm there is no bug in it. After this simple part is okay, you can then add more elements and test again. Repeat this add-and-test process until the complete formula is built.

© Welkin Systems Limited

# Part

# 10 Printing and Creating Reports

RaceMate comes with a simple-to-use yet powerful report writer. After you have designed a view, you may create a report for printing out the view by only a couple of mouse clicks.

You may design as many reports as you like for a single view. Especially if your view is very large and contains many columns, you may create several reports each printing out a different set of columns.

**>>> Subtopic of this section <<<**

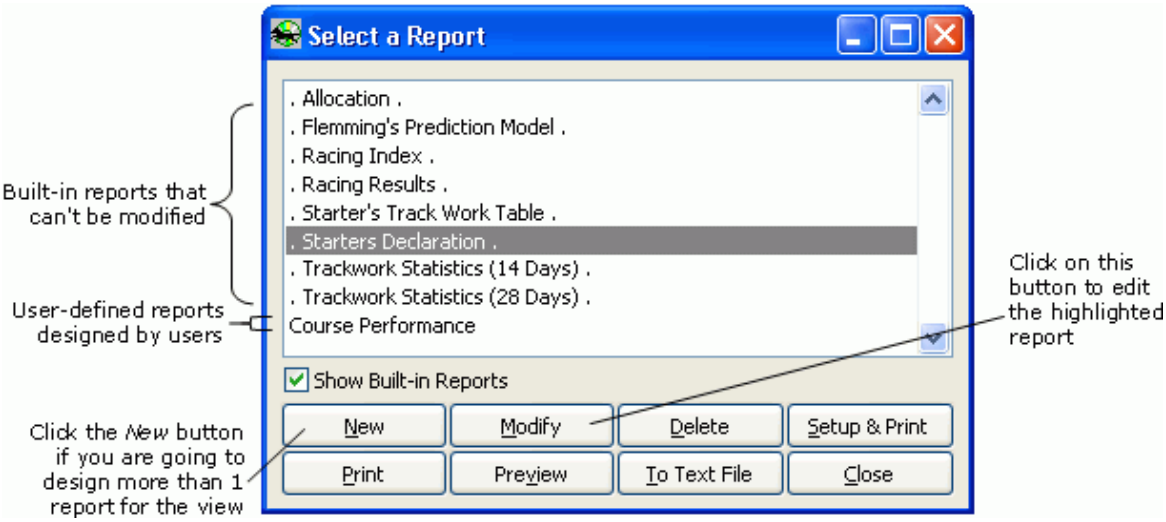
- Creating a Report for a View
- Report Designer

## 10.1 Creating a Report for a View

Follow the example steps below to create a new report:

1. In the Starters Views page of the Race Data form, switch to the Course Performance view.
2. Click on Print button on the RaceMate Toolbar.

The Report List will appear and the report for the current view is highlighted.


Select a Report
. Allocation .
. Flemming's Prediction Model .
. Racing Index .
. Racing Results .
. Starter's Track Work Table .
. Starters Declaration .
. Trackwork Statistics (14 Days) .
. Trackwork Statistics (28 Days) .
Course Performance
Show Built-in Reports
New
Modify
Delete
Setup & Print
Print
Preview
To Text File
Close
Built-in reports that can't be modified
User-defined reports designed by users
Click the New button if you are going to design more than 1 report for the view
Click on this button to edit the highlighted report

In the list, you can see two types of reports: Built-In Reports (names begin with a ".") and User-Defined Reports. Built-In Reports come with RaceMate and cannot be modified. User-Defined Reports must be designed by users before they can be used.

3. Click on the Modify button to show the report designer so that you can prepare the report for use. (Please note that if you click on the Preview button instead of the Modify button, the report designer will be show as well because a user-defined

© Welkin Systems Limited

report must be edited at least once before it can be printed or previewed.) In the RaceMate report designer shown in the following figure, a Column Preview Table is at the upper part. The table gives you handy information about:

- Column width
- Gap width between columns
- Total column width
- Column heading

At the left of the designer, under the Column Preview Table, a Column Information list box is found. Click on Horse in the list to highlight the column. Attributes of the column are show in the Column Attributes Edit Area.

4. Change the width of the Horse column to 6 cm by clicking on the Width Spinner. At the same the, you will see the width of the Horse column changes in the Column Preview Table and the value of the Remaining width changes.

   Note: If the Remaining width drops to zero, the report columns just occupy the whole report width (excluding the left and right margins)

5. Click on Save+Exit to close the Report Designer and return back to the Select a Report dialog.

## 10.2 Report Designer

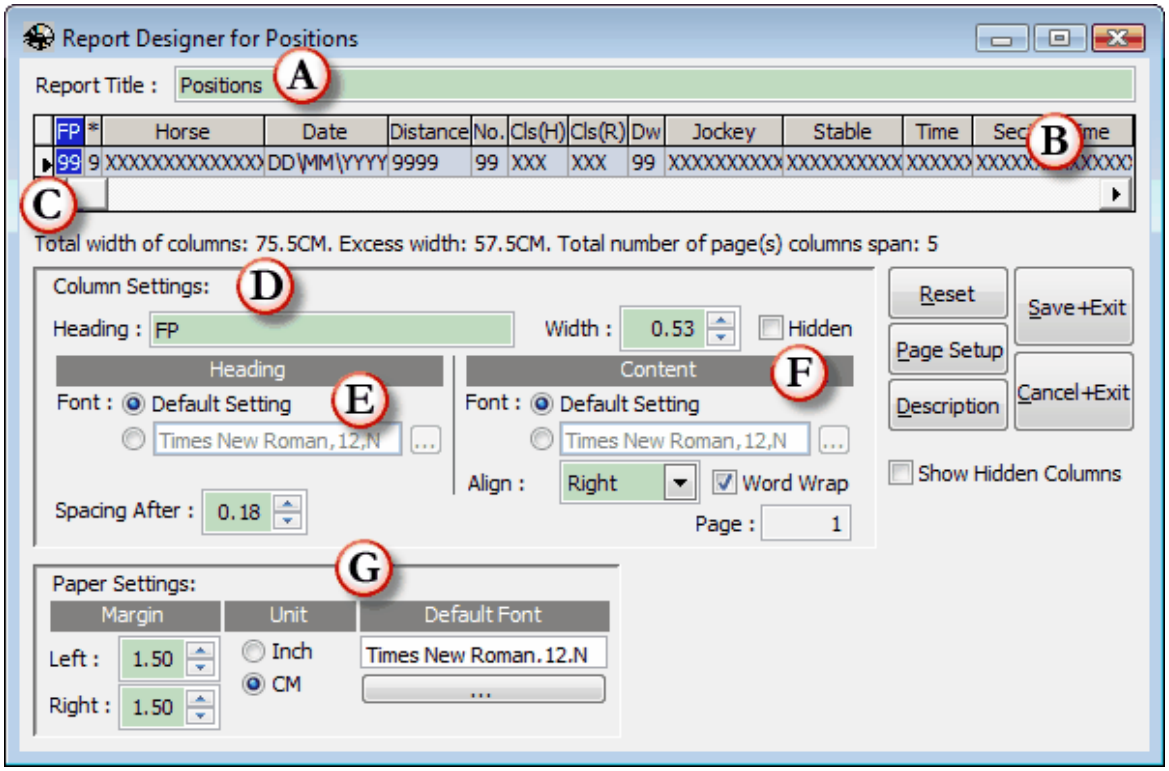
There are 2 types of reports:

1. Built-in reports (a dot in front of the report title)
2. User-defined reports

You are not allowed to modify a built-in report, but you can create user-defined reports for the same view.

Click on the [Print] button or select [Print] from the [File] menu. The click on [New] to create a new report or [Modify] to modify the existing report. You will then see the following form. The function of each part is as follows.

© Welkin Systems Limited




**A. Report Title**

The title of the report. You can modify it.

**B. Preview of Report Columns**

This area show how the data are shown in the report. Resize the columns with the mouse will directly affect how the report prints.

**C. Column Width Info Bar**

Shows total number of pages and other relevant information.

**D. Column Settings**

Shows the name and width of the highlighted column. You can make a column disappear from the report by checking the [Hidden] checkbox.

**E. Heading**

Font for column heading and space after column (between this and next columns)

**F. Content**

Font, word wrap, and alignment of the column. The number in the [Page] box shows on which page the column will be. Normally there will only be one page.

**G. Paper Settings**

Margins of the report and default fonts.

© Welkin Systems Limited

You will find the following buttons on the form.

**Reset**
Cancel all changes and restore what you see when this form is open.

**Page Setup**
Define attributes of print out, e.g, paper size, printer, etc.

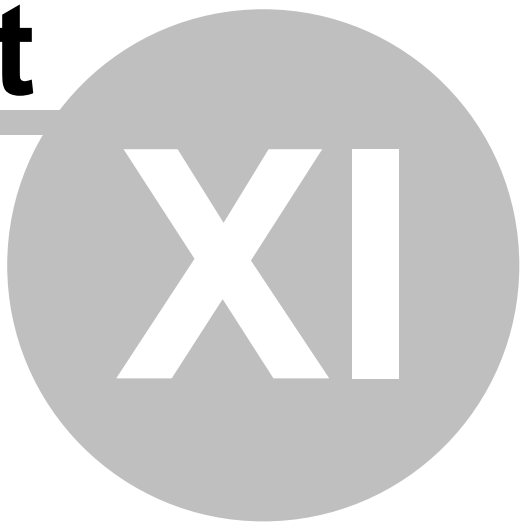
**Description**
Report description

**Save+Exit**
Save change and close the form

**Cancel+Exit**
Cancel change and close the form

© Welkin Systems Limited

# Part



# 11 Multimedia Resources

You will find a "Links" page in the main data forms of RaceMate. In the page, you will see a set of autolinks linking to the official multimedia files of the Jockey Club. You may also add your own multimedia links into the page.

**>>> Subtopics of this section <<<**

- Autolinks
- Managing Your Own Multimedia Resources
- Saving Web Video/Photo into Disk

**See also:**

Autolinks | Managing Your Own Multimedia Resources

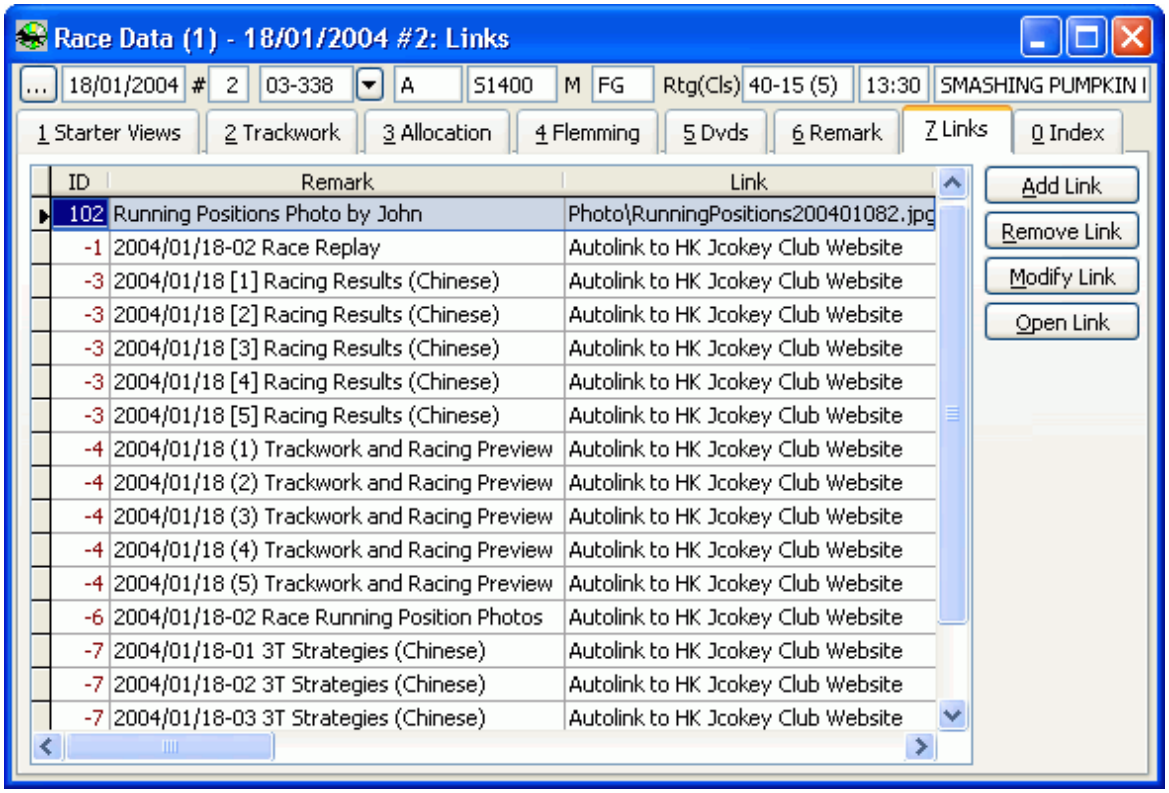
## 11.1 Autolinks

You will see a list of links in the link page. Links with a negative ID are autolinks inserted by RaceMate automatically. You can double-click on a link (or click on the [Open Link] button) to open, or play, the linked multimedia file.

Autolinks are generated by RaceMate based on a set of rules. The existence of a link in the list does not guarantee the link can be open. Whether the link is functional is subject to the whether the file is uploaded onto the server by the Jockey Club. If you failed in opening a link, you may try again later.

In case you find some of the autolinks not useful, you may remove them from the list by opening the Options and System Information and unchecking corresponding checkboxes.

© Welkin Systems Limited

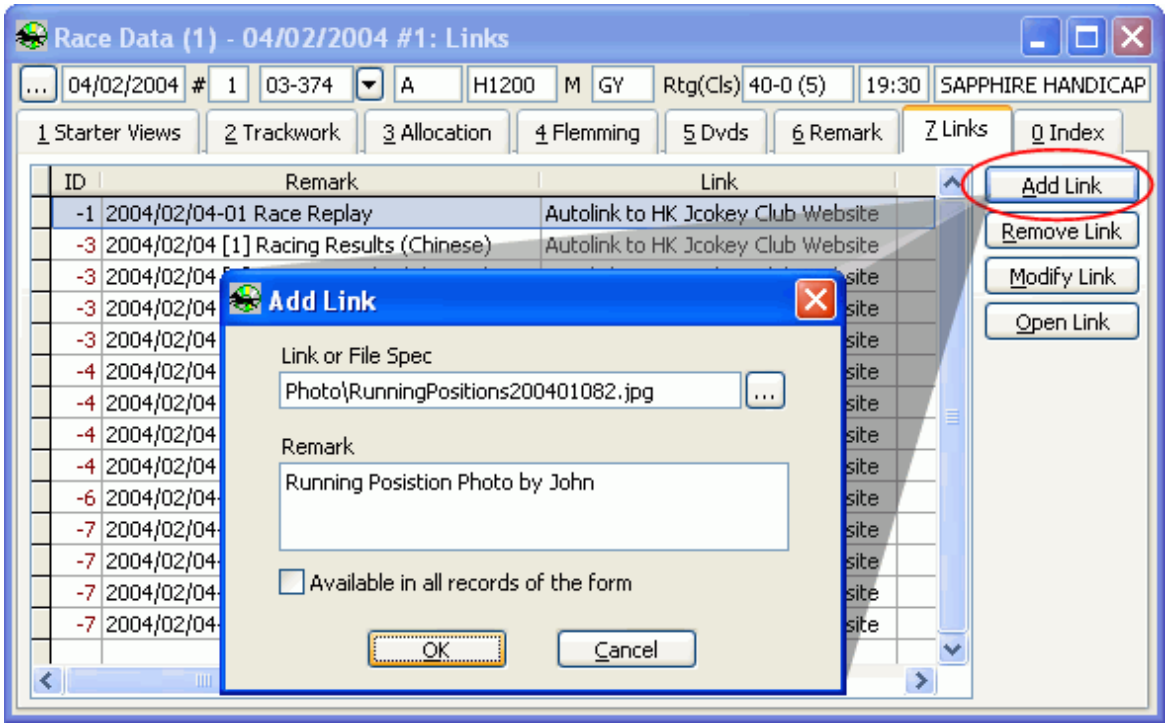



| ID | Remark | Link |
|---|---|---|
| 102 | Running Positions Photo by John | Photo\RunningPositions200401082.jpg |
| -1 | 2004/01/18-02 Race Replay | Autolink to HK Jcokey Club Website |
| -3 | 2004/01/18 [1] Racing Results (Chinese) | Autolink to HK Jcokey Club Website |
| -3 | 2004/01/18 [2] Racing Results (Chinese) | Autolink to HK Jcokey Club Website |
| -3 | 2004/01/18 [3] Racing Results (Chinese) | Autolink to HK Jcokey Club Website |
| -3 | 2004/01/18 [4] Racing Results (Chinese) | Autolink to HK Jcokey Club Website |
| -3 | 2004/01/18 [5] Racing Results (Chinese) | Autolink to HK Jcokey Club Website |
| -4 | 2004/01/18 (1) Trackwork and Racing Preview | Autolink to HK Jcokey Club Website |
| -4 | 2004/01/18 (2) Trackwork and Racing Preview | Autolink to HK Jcokey Club Website |
| -4 | 2004/01/18 (3) Trackwork and Racing Preview | Autolink to HK Jcokey Club Website |
| -4 | 2004/01/18 (4) Trackwork and Racing Preview | Autolink to HK Jcokey Club Website |
| -4 | 2004/01/18 (5) Trackwork and Racing Preview | Autolink to HK Jcokey Club Website |
| -6 | 2004/01/18-02 Race Running Position Photos | Autolink to HK Jcokey Club Website |
| -7 | 2004/01/18-01 3T Strategies (Chinese) | Autolink to HK Jcokey Club Website |
| -7 | 2004/01/18-02 3T Strategies (Chinese) | Autolink to HK Jcokey Club Website |
| -7 | 2004/01/18-03 3T Strategies (Chinese) | Autolink to HK Jcokey Club Website |

## 11.2 Managing Your Own Multimedia Resources

You may insert links pointing to your own resources. Actually, the files are not limited to audiovisual types. Any file types which are properly registered in Windows are valid candidates. For example, you may add Microsoft Word and Excel files, or website addresses. The following figure shows the Links page of the Race Data form

© Welkin Systems Limited




### Buttons for Managing Links

- Add Link: Add a new link into the links list
- Remove Link: Remove a link from the links list
- Modify Link: Modify the content of a link
- Open Link: Open the file pointed by the link

### The Add Link and Edit Link Forms

When you click on the [Add Link] button or the [Modify Link] button, RaceMate will show the "Add Link" or "Modify Link" form. These two forms are exactly the same except that the captions in the title bar are different. The main elements of the form are listed below:

- Link or File Spec edit box
  Enter the path of the file (including the file name) to be linked here. You may click on the small button on the right of the edit box to browse files in the disk. When you select a file, RaceMate will put the path of the file into the edit box. You may enter a full path or a path relative to the home folder of RaceMate.
- Remark edit box
  A sentence describing the link
- Available in all records of the form check box
  When this option is selected, the link will appear in all records of the form (i.e., all races in the Race Data form or all horses in the Horse form). Otherwise the link will only be show in the current record.

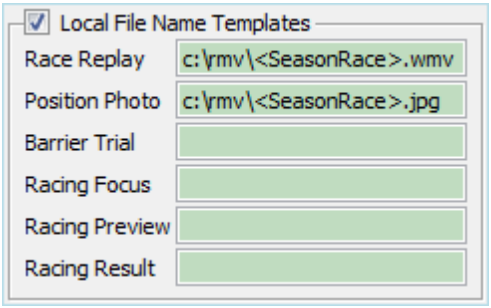
© Welkin Systems Limited

## 11.3 Saving Web Video/Photo into Disk

### Transferring Multimedia Files from Websites into Local Drive

Many users like downloading video files and position pictures from websites into local hard disk to get fast playing and viewing. You can then keep your files even if they are removed from the web. If you are doing similar things, you no longer need to manually create links to the files that you saved.

You can open the "Options and System Information" form from the [Tools] menu. At the right top of the form you will see an area for entering a template for each file type as follows. Once you correctly enter corresponding templates into the boxes in the figure below (you can leave some of them blank), RaceMate's default links or buttons will point to these local files rather than the one on websites.




| Template Element | Meaning |
|---|---|
| <Year> | 4-digit year |
| <Year-> | 2-digit year (07 for 2007) |
| <Month> | 2-digit month (03 for March) |
| <Month-> | Month (3 for March) |
| <Day> | 2-digit day (03 for the 3rd day of a month) |
| <Day-> | day (3 for the 3rd day of a month) |
| <RaceNo> | 2-digit race number (02 for the 2nd race) |
| <RaceNo-> | Race number (2 for the 2nd race) |
| <SeasonRace> | 20080073 for season 2008-2009 race 73 |
| <SeasonRace-> | 080073 for season 2008-2009 race 73 |
| <DateStr> | 20080301 for 1 March 2008 |
| <DateStr-> | 080301 for 1 March 2008 |

### Expressions Or Non-Expressions

The screen capture above shows 2 real hard disk paths. Besides real paths you can put a "=" at the start position of the text box to turn the string into an expression. The double quote (") in the example below is needed because there is a space in the folder name "my video".

```
='"c:\my videos\'+Left(<SeasonRace>,4)+'-'+Right(<SeasonRace>,3)+'.wmv"'
```

If you view race 331 of season 2008-2009, the above expression will be evaluated as follows.

```
='"c:\my videos\'+Left("20080331",4)+'-'+Right("20080331",3)+'.wmv"'
```

Finally we will get "c:\my videos\2008-331.wmv"

© Welkin Systems Limited

**Examples**

| Field | Template | Click On | File Opened |
|---|---|---|---|
| Race Replay | c:\rmv\<SeasonRace>.wmv | Race Replay link or button of race 7 on 29/4/2007 (season 06-07) | c:\rmv\20080559.wmv |
| | c:\Replay\<SeasonRace->.ivr | | c:\Replay\060559.ivr |
| | c:\File\HK<Year-><Month><Day>_<RaceNo>.ivr | | c:\File\HK070429_07.ivr |
| | c:\Video\HK-<DateStr->-<RaceNo>.ivr | | c:\Video\HK-070429-07.ivr |
| | ='"c:\my videos\'+Left(<SeasonRace>,4)+'-'+Right(<SeasonRace>,3)+'.wmv"' | | "c:\my videos\2008-559.wmv" |
| Position Photo | c:\Photo\<SeasonRace>.jpg | Positions Photo link of race 7 on 29/4/2007 | c:\Photo\20060559.jpg |

".wmv" and ".ivr" are the extension of Windows Media Player's and RealPlayer's files respectively. If you don't have RealPlayer installed you will not be able to play the .ivr files. By default Windows Media Player is installed in every Windows.

© Welkin Systems Limited

**How can I save a web video as a local file?**

There are many tools which help you do the job. For example the latest version of RealPlayer and FlashGet. Both of them are free of charge. Please read their manuals for detailed operations.

© Welkin Systems Limited

# Part

# 12 Statistics and Data Analysis

Statistics features of RaceMate are integrated with many forms. For example,
- Statistics pages are found into the Horse, Stable, and Jockey forms
- Statistical columns are put into starter views

You can also insert your own statistical columns into views.
Beside such integrated statistics features, RaceMate provide with users the following professional tools for carrying out professional statistics:

1. The Quick Statistics Form

The Quick Statistics Form is a dropdown menu based interface for carrying out general statistics. Everybody, including the laypeople, will find it easy to use. Further more, it takes advantage of the famous Rushmore technology of Microsoft Visual FoxPro and provides us with an extremely fast calculation speed. However, its criteria setting capability is limited by the simple user interface. If you find that your criteria are difficult to be put into the form, you may turn to the much more powerful tool: Race Query and Comprehensive Statistics Form.

2. Race Query and Comprehensive Statistics Form

The "Race Query and Comprehensive Statistics Form" can do all the statistics the Quick Statistics Form does and much more. However the calculation speed of it is lower than the Quick Statistics Form. Besides statistics, it is an excellent tool for listing a set of races which satisfies the criteria that you specified. If you find that even the "Race Query and Comprehensive Statistics Form" is not able to do what you want, you may turn to another professional tool: Starter View Model Statistics.

3. Advanced Statistics via View Models

"View Model Statistics" is the most powerful statistics tool of RaceMate. We can formulate statistics criteria into MCL functions and construct a view model. The statistics function of the view model will solve most statistics problem. MCL provides the model with tremendous extensibility which is limited only by your imagination.

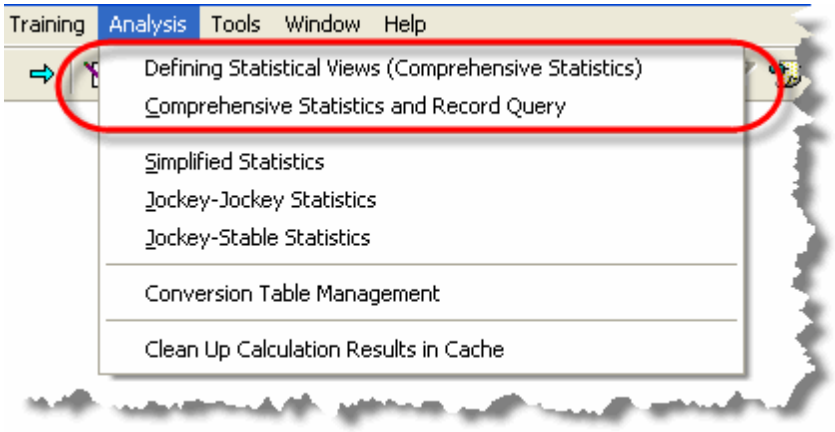
**>>> Subtopics of this section <<<**


- Type 1: Comprehensive Statistics & Record Query
- Type 2: Statistics via Starter Views (Race Data Form)
- Type 3: Simplified Statistics Form


## 12.1 Type 1: Comprehensive Statistics & Record Query

Choose [Defining Statistical Views (Comprehensive Statistics)] from the [Analysis] menu to define criteria of statistics and queries. The [Comprehensive Statistics and Records Query] menu opens a form the display the results getting by running the criteria and queries that you defined.

© Welkin Systems Limited




See the Type 2: Histroy and Statistics View section for more information.

**>>> Subtopics of this section <<<**

- Defining Structure of Statistics Views and Queries
- Statistical View and Record Filtering Results
- Editing View Particulars
- Editing View Column/Criteria Items
- Query Operators for Requirements
- Creating and Removing OR or AND Criteria Group
- Data Period and Display Mode
- Statistical Items - the [Stat] Column
- Viewing Past Records and Move Columns in Stat View Page

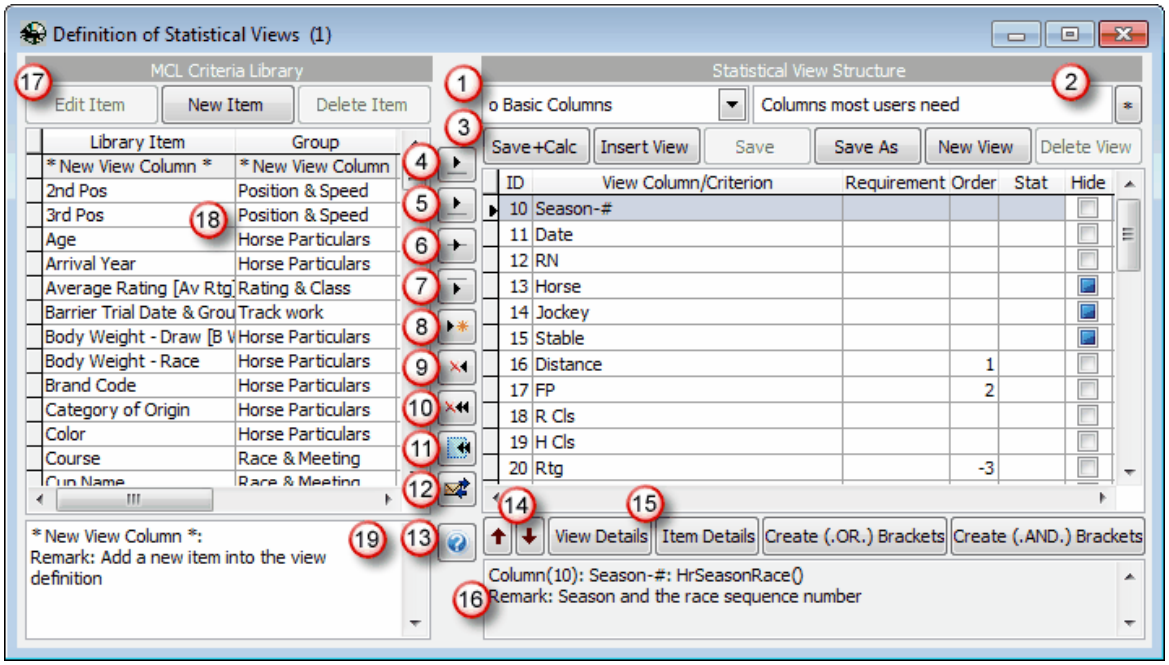
### 12.1.1 Defining Structure of Statistics Views and Queries

If you want to see a list of requirement operators, please see the following page:

Query Operators for Requirements

When you choose [Defining Statistical Views (Comprehensive Statistics)] from the [Analysis] menu, you will see the following form. You can change the structure of a view or create new views on this form. However please note that any changes made to a system built-in view will be lost when you restart RaceMate.

© Welkin Systems Limited




The above form is composed of the following parts:

**View Title (1) - (2)**

On the top of the form, you can find a dropdown list (1) showing the title of the currently editing view. You can change current view by pulling down the list and selecting another view. At the right hand side of the list, there is a textbox (2) showing a description about the view. You may click on the [*] button on its right to zoom the description on a form.

**Buttons for Creating, Deleting and Running Views (3)**

Below the view title list, you can see a row of buttons for creating, deleting and running views:

[Save+Calc]

Click on this button to open the view on the "Comprehensive Statistics and Record Query" form.

[Insert View]

Click on this button to choose a view from a list and insert it above or below the currently highlighted row in the criteria table. You can save a number of frequently needed columns into particular views and insert them into your new views on-the-fly. In the View Details editing form, you can hide these particular views from the view list of data forms.

[Save]

Save current change and continue editing.

[Save As]

© Welkin Systems Limited

Save the structure currently editing into a new view.

[New View]

Create a blank new view.

[Delete View]

Delete the view currently editing.

**View Columns/Criteria Maintenance Buttons (4) - (12)**

[Append to End] (4)

Add item from MCL function library into bottom of view structure table

[Insert below Current] (5)

Insert item from MCL function library into view structure criteria table just below the current item

[Overwrite Current] (6)

Copy item from MCL function library into view structure table overwriting the current one

[Insert above Current] (7)

Insert item from MCL function library into view structure table just above the current one

[Insert New View Column (Blank) above Current (8)

Insert blank view item into view structure table just above the current one for editing

[Remove Current] (9)

Remove item currently highlighted from view structure table

[Remove All] (10)

Remove all items from view structure table

[Reset View Structure to Default] (11)

Restore default structure of built-in views. RaceMate will automatically restore all built-in views into their default structure every time you restart RaceMate. However if you want to reset a built-in view without restarting RaceMate, you may click on this button.

[Export & Import View Structure] (12)

Export or import view structure to or from file

© Welkin Systems Limited

**Help Button (13)**

Push this button to show the help screen

**Repositioning View Items (14)**

[Move Item Up]

Click on this button to move the currently highlighted criteria up in the criteria table.

[Move Item Down]

Click on this button to move the currently highlighted criteria down in the criteria table.

**View Particulars Fine-Tuning Buttons (15)**

[View Details]

Click on this button to show the View Details form, on which you can change the view title, define the forms that will show the view, etc. See EditingView Particulars for details.

[Item Details]

Click on this button to show the "Edit View Item" form, on which you can fine tuning the properties of a view item. See Editing View Column/Criteria Items for details.

[Create (.OR.) Brackets]

Click on this button to put the currently highlighted view item into an "OR" criteria group. An OR criteria group is a group of criteria, which are indented in the view structure editing area. If any criteria in this group is valid, the whole OR criteria group is regarded as valid. See Creating and Removing OR or AND Criteria Group for details.

[Create (.AND.) Brackets]

Click on this button to put the currently highlighted view item into an "AND" criteria group. An AND criteria group is a group of criteria, which are indented in the view structure editing area. Only if all criteria in this group are valid, the whole AND criteria group is regarded as valid. See Creating and Removing OR or AND Criteria Group for details.

**Column Information Box**

A text area at the right bottom of the form shows the details about the column.

**MCL Criteria Library Item Editing Buttons (17)**

[Edit Item]

Push this button to edit a user-defined library item

[New Item]

© Welkin Systems Limited

Push this button to create a new user-defined item and put it into the MCL Criteria Library

[Delete Item]

Push this button to delete a user-defined library item

**MCL Criteria Library Items (18)**

The left table shows all the items in the MCL criteria library. You may highlight any of the item and push buttons (4) to (8) to move it into the view structure table.

**MCL Criteria Library Item Information (19)**

A text area at the left bottom of the form shows the details about the highlighted MCL criteria item.

Please see Statistics and Data Analysis for more information.

### 12.1.2 Statistical View and Record Filtering Results

You can show the results of your view criteria definition in a Statistical View page. This page can be found in the following forms:

- Comprehensive Statistics & Records Views Form
- Horse Data Form
- Jockey Data Form
- Stable Data Form

**Display Modes of a View**

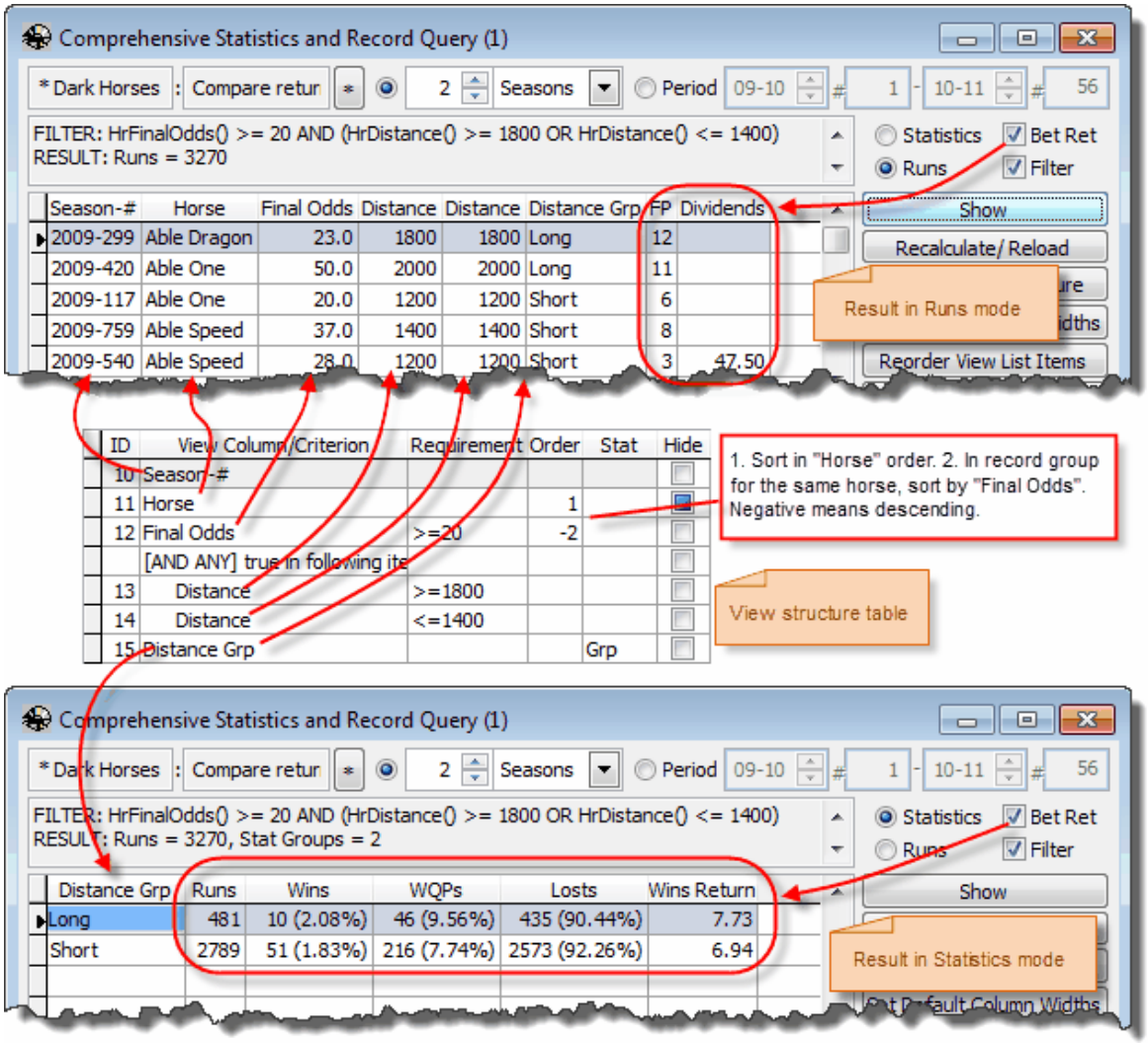
Results can be shown in 2 different view modes: Runs (upper form in the figure below) and Statistics (lower form of the figure below). When the view is open, RaceMate firstly retrieves all records which satisfy the requirements define in the view structure table. Each row in the view structure table matches a field of the records. The Runs mode simply shows this set of retrieved records. When Statistics mode is chosen, RaceMate uses this set of records to carry out grouping and other statistics actions according to the setting of the "Stat" column of the view structure table. The final grouping and statistics are then shown in the "Statistics" mode.

**Result Sorting**

- You can specify how the result is sorted. Put "1" into the [Order] column to sort the result by the underlying item. When there are identical item values, RaceMate will sort the these items by the item with "2" in the [Order] column, and so on.
- Change the [Order] value to negative to indicate a descending sort.

In the example in the figure below, RaceMate will firstly sort the result in Horse order. In record group for the same horse, records will be sorted by "Final Odds" in descending order.

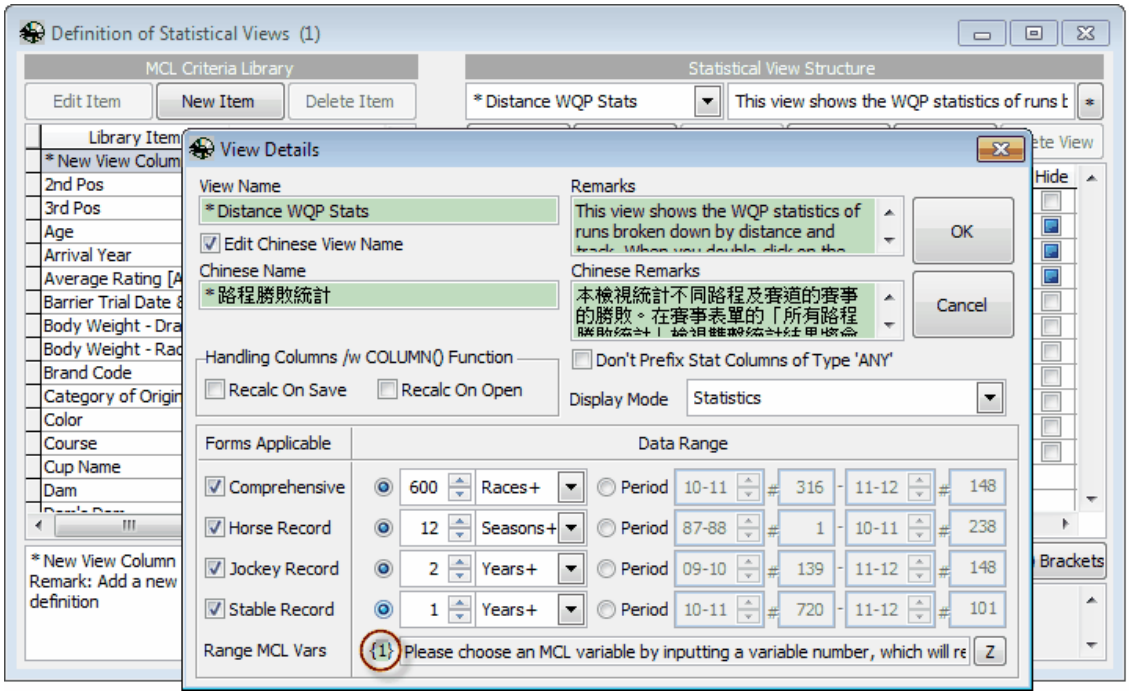
© Welkin Systems Limited


Comprehensive Statistics and Record Query (1)
FILTER: HrFinalOdds() >= 20 AND (HrDistance() >= 1800 OR HrDistance() <= 1400)
RESULT: Runs = 3270
Result in Runs mode
1. Sort in "Horse" order. 2. In record group for the same horse, sort by "Final Odds". Negative means descending.
View structure table
FILTER: HrFinalOdds() >= 20 AND (HrDistance() >= 1800 OR HrDistance() <= 1400)
RESULT: Runs = 3270, Stat Groups = 2
Result in Statistics mode

## 12.1.3 Editing View Particulars

When you click on the [View Details] button or the [New View] button on the "Definition of Statistical Views" form, you will see the following form. You can edit properties of a view on the form. However please note that any changes (except the data ranges) made to a system built-in view will be lost when you restart RaceMate.

© Welkin Systems Limited




You may carry out the following tasks on the form:

**Edit View Title and Remarks**

In the boxes at the top of the form you may rename the view and change the remarks for the view. If you have check the [Edit Chinese View Name] checkbox, you will see four boxes for Chinese and English view names and remarks respectively. Otherwise you will see only the two for English modes.

**Handling Columns /w COLUMN Function**

[Recalc On Save]

If you have make changes to an input column, which is referred by a "Column( )" function in another calculated column, you may want to update the calculated column to reflect the latest changes you have made when you save your changes. Check the [Recalc On Save] checkbox will perform a recalculation every time you save your view after inputting. However, if you are sure that there is no calculated column referring to an input column, you should uncheck this box to speed up saving.

[Recalc On Open]

Sometimes input columns will be updated in other views (linked column) rather than in this view. If there is any calculated column which is referring to this input column by using a "Column( )", and you want to update the column to reflect the changes made in other views every time you open the view, check the [Recalc On Open] checkbox. However, if you are sure that there is no calculated column referring to an input column, you should uncheck this box to speed up the view opening speed.

**Showing Statistics Results**

[Don't Prefix Stat Columns of Type 'ANY']

If you have defined a column of the Stat Type: ANY (rather than MAX or MIN), race will return the first value it encounters in the group from the database and prefix the column heading with "any:". If you don't want the prefix, please check [Don't Prefix

© Welkin Systems Limited

Stat Columns of Type 'ANY']

[Display Mode]

1. Races

   List all the races which satisfies the duration and criteria settings

2. Races+Returns

   List all the races which satisfies the duration and criteria settings to gather with finishing positions and the dividends of winning horses

3. Statistics

   List the statistics figures according to the grouping and statistics type defined in the STAT column of the view definition. If nothing has been defined in the STAT column, all races satisfying the criteria will be regarded as a single group

4. Statistics+Returns

   Display the same columns of the option above together with columns showing win counts and betting returns

**Data Ranges**

The lower part of the form defines the data range either to be displayed or to be used for statistics. Since part records involved by different entities are different, you may define different data ranges for different form types. Please see Setting Data Range in an Argument about how to define data range via passing arguments.

**Removing Current View from View List of a Particular Form**

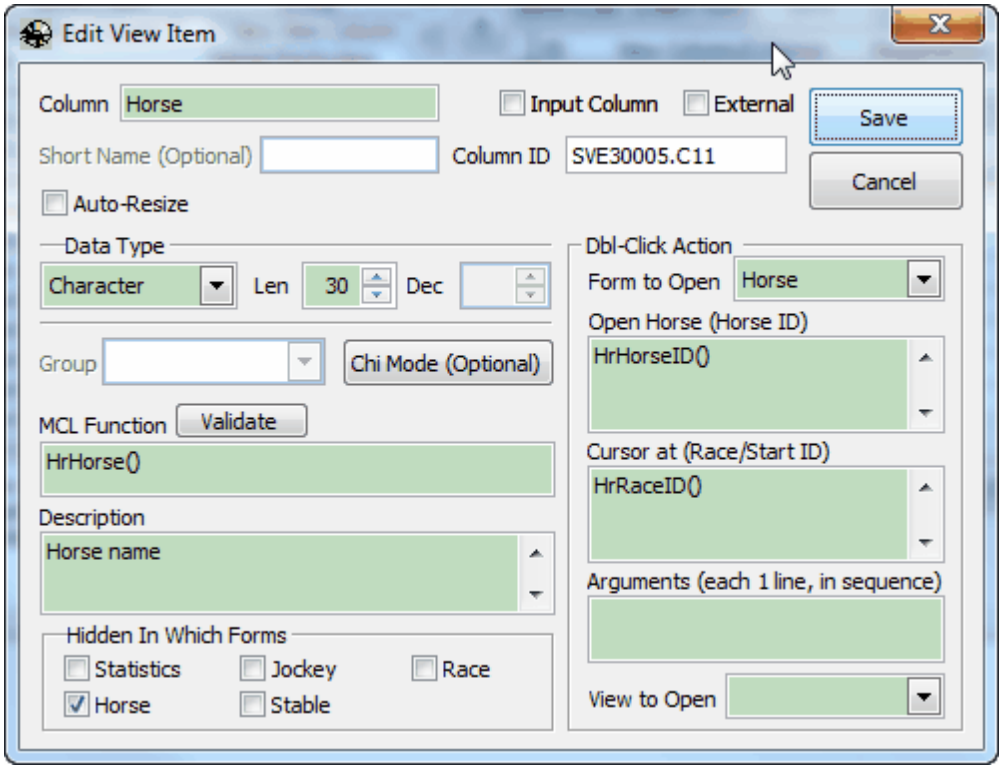
In case the current view is not applicable for a particular form, you may uncheck the corresponding checkbox at the left bottom.

### 12.1.4 Editing View Column/Criteria Items

On the "Definition of Statistical Views" form, when you carry out any of the following actions, you will see the view criteria property editing form as shown in the figure below:

- Click on the [Item Details] button below the view structure grid. (Form title will be shown as "Edit View Item")
- Move [* New View Column *] from the MCL Function Library table into the Statistical View Structure table. (Form title will be shown as "Add New View Item")
- Click on the [New Item] button at the top of the MCL Function Library grid. (Form title will be shown as "Add New Library Item")
- Click on the [Edit Item] button at the top of the MCL Function Library grid. Please note that only user-defined library items are editable. (Form title will be shown as "Edit Library Item")

© Welkin Systems Limited




Fields on the forms are explained as follows:

[Column]

Name of column

[Short Name]

This field is only applicable to library items. When a library item is moved into the view structure grid, the system will use this name, rather than the full name, as column heading.

[Input Column]

Check this box if the column is used for inputting user ratings or comments. You can input an MCL function for an input column. However, this MCL function will be run only once the first time results are shown. You may use this MCL function to populate the field with a default value.

[External]

- If this box is checked, RaceMate will fill this column with the data of another (external) view. You must enter the column ID of this external column into the Column ID box below, which you may copy and paste by opening the external view and edit the target column. Please note that RaceMate will not validate the Column ID that you input. You are responsible for ensuring its correctness.
- For an External column, what you enter into the MCL Function box will be irrelevant, since it always replicates the values of the external column. Please leave the MCL function box blank.
- If the External column is for inputting your ratings or comments, you can make it

© Welkin Systems Limited

editable by checking [Input Column] at the left hand side. Then ratings or comments can be input in either the current view or the external view.

[Auto-Resize]

When result form is resized, column width changes accordingly.

[Column ID]

Column ID will automatically be generated by the system except that this is an external column.

[Data Type]

In the three boxes, please input correct data type, length (number of characters or digits including the decimal dot and figures after the dot), and decimal width for the column.

[MCL Function]

MCL function returning a value for the column

[Description]

Optional descriptive text for the column

[Dbl-Click Action]

Choose a form to be opened when the column is double-clicked. In the box below, you must enter a function returning a key ID, which determines which record is to be shown in the target form. In another box below, you can further input another optional function, which returns another key ID for locating a particular line in the target form.

[Hidden In Which Forms]

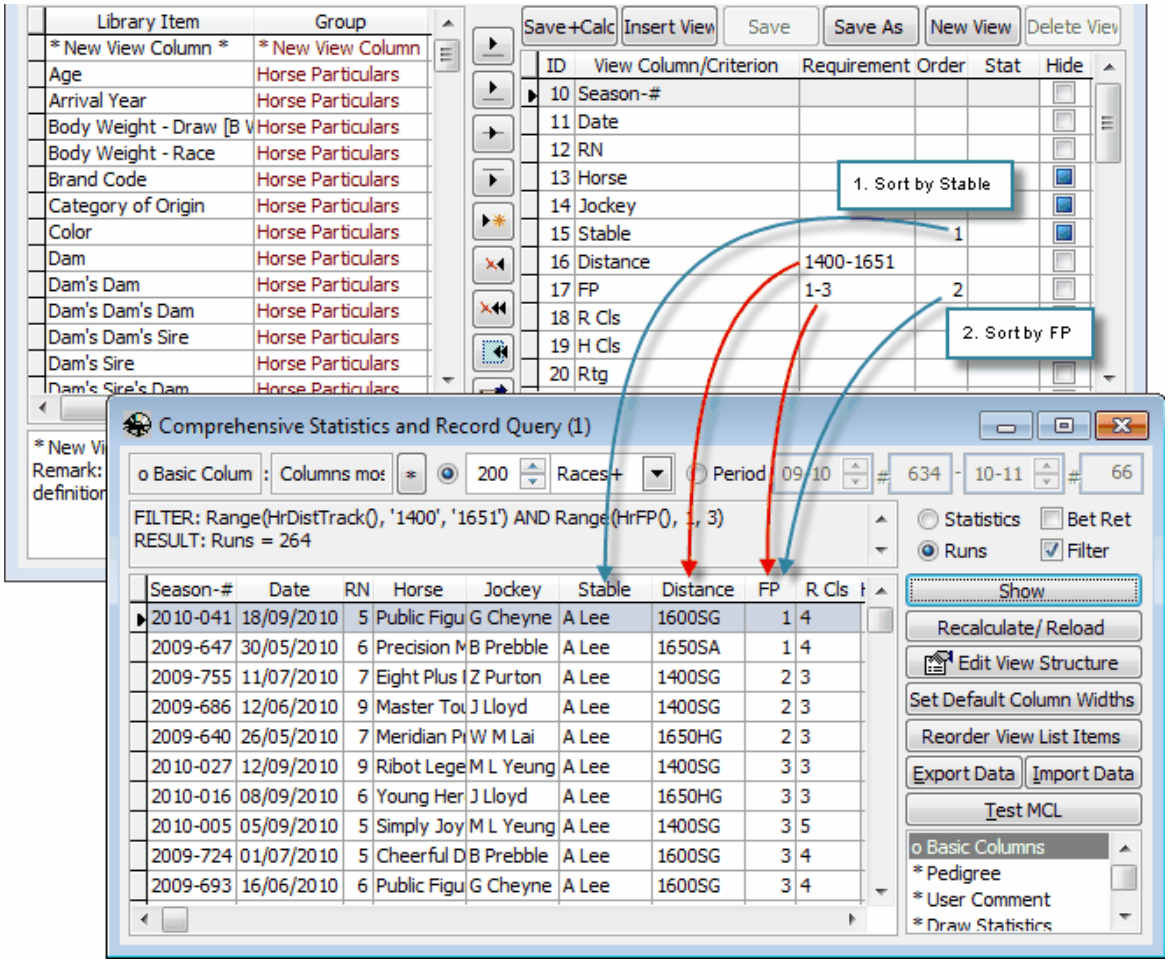
If the data of this column is irrelevant for certain forms, you may check those forms in the area hidden the column in the forms.

### 12.1.5 Query Operators for Requirements

You may set criteria using the following operators. Please note the following rules for the requirement text:

1. All operators except "-" must be put at the beginning of the requirement text.
2. Quotes of character strings and date values can be omitted. However if omitting quotes will result in confusion, please keep them.
3. By default, text comparing and matching is case-insensitive. If you want RaceMate to do a case-sensitive compare, please put a case-sensitive modifier "^" at very beginning of the requirement text.
4. "!" means "NOT", which is a modifier to be put at the beginning of a requirement.
5. You may not input any formula or equation into the requirement field.

© Welkin Systems Limited




Criteria Operators

| Operator | Meaning | Examples |
|---|---|---|
| = | Exact Match: Cell value is exactly equal to...<br>(Trailing space is insignificant) | =Good<br>='Good'<br>=3 |
| > | Greater: Cell value is greater than...<br>(Dictionary order. Trailing space is insignificant) | >ABC<br>>'ABC'<br>>3 |
| < | Smaller: Cell value is smaller than...<br>(Dictionary order. Trailing space is insignificant) | <ABC<br><'ABC'<br><3 |
| >= | Greater or equal: Cell value is greater than or equal to<br>(Dictionary order. Trailing space is insignificant) | >=ABC<br>>='ABC'<br>>=3 |
| <= | Smaller or equal: Cell value is smaller than or equal to<br>(Dictionary order. Trailing space is insignificant) | <=ABC<br><='ABC'<br><=3 |
| .. - .. | Range: Cell value is greater than or equal to the value at the left hand side of "-" and smaller than or equal to the value at the right hand side of the "-"<br>(Dictionary order. Trailing space is insignificant) | A – D<br>'A' – 'D'<br>1 - 3<br>{^2007/3/1} - {^2007/10/31} |

© Welkin Systems Limited

| | | |
|---|---|---|
| <> OR # | Unequal: Cell value is not equal to...<br>('ABC' is not equal to 'AB'. Trailing space is insignificant) | <>ABC<br><>'ABC'<br><>13 |
| $ | Function value found in requirement: Cell values is found inside the requirement string<br>(Only applicable to character or text data) | $ABC<br>$'ABC' |
| Unquoted requirement with wildcard | Wildcard expression: Cell value matches a wildcard (* or ?) pattern<br>(Only for character or text data. Not applicable to strings quoted with " ", ' ', or [ ]) | W*t?<br>Wh?te |
| (no any of above operators) | Automatic: If no operator is put before the value, RaceMate will create filter expressions according to the following rules:<br>For columns of non-character data types:<br>• For Logical type, a requirement begins with any of the following characters will be regarded as False: . F., F, 0, !, N<br>• For columns of any data type other than Character or Logical, the "=" operator will be applied<br><br>For columns of character type:<br>• If the requirement is a quoted string, the operator "=" will be applied<br>• For an unquoted requirement without any wildcard character, a "*" wildcard character will automatically be appended to the end of the requirement. The requirement will then be treated as a wildcard expression. | |

**Exact-Case Modifier**

Precede the requirement with a caret mark (^) requires strings to be compared in exact-cases ('Whyte' is not equal to 'whyte').

**NOT Modifier**

Precede the requirement with an exclamation mark (!) means "NOT" (!'Whyte' means #'Whyte').

**Examples**

| MCL in Column | Requirement | Meaning | Data Type of MCL |
|---|---|---|---|
| HrDistance( ) | 1400 | Filter all 1400M races | Numeric |
| HrDistance( ) | =1400 | Filter all 1400M races | Numeric |
| HrDistance( ) | <>1400 | Exclude all 1400M races | Numeric |
| HrTrack( ) | 'A+3' | Filter races of A+3 track | Character |
| HrTrack( ) | 'A' | Filter races of A track. A+3 track will not be included. | Character |
| HrTrack( ) | A* | Filter races of A track including A, A+2, A+3, etc | Character |
| HrTrack( ) | 'A*' | No race will be shown because * in quotes will not be regarded as a | Character |

© Welkin Systems Limited

| | | | |
|---|---|---|---|
| | | wildcard | |
| HrTrack( ) | a+? | Filter races of A+.. track. A will not be included. String matching is case insensitive by default. | Character |
| HrTrack( ) | !A* | Show races of all tracks other than A, A+2, A+3, etc | Character |
| HrTrack( ) | !A | Identical to above example because the system will automatically append an * | Character |
| HrTrack( ) | =A* | No race will be shown. Once the requirement begins with an operator, * will not be a required character rather than a wildcard. | Character |
| HrTrack( ) | $AC | Filter races of A and C tracks. A+2, A+3, etc will not be included | Character |
| HrDate( ) | 13/6/2007 | Filter races on 13/6/2007. If date format on your screen is mm/dd/yyyy, you must input 6/13/2007 rather than 13/6/2007. | Date |
| HrDate( ) | {^2007/6/13} | Filter races on 13/6/2007. If you formally quote a date, it must be in {^yyyy/mm/dd} format. | Date |
| HrDate( ) | >13/6/2007 | Filter races after 13/6/2007 | Date |
| HrDate( ) | 13/6/2007-15/9/2007 | Filter races during 13/6/2007-15/9/2007 | Date |
| IsTurf( ) | Y | Filter all turf course races | Logical |
| IsTurf( ) | 1 | Filter all turf course races | Logical |
| IsTurf( ) | N | Filter all non-turf course races | Logical |
| IsTurf( ) | 0 | Filter all non-turf course races | Logical |
| HrRunning( ) | *slow* | Filter all starts with the word 'slow' found in JC running comment | Text |
| HrRunning( ) | !*slow* | Filter all starts without the word 'slow' found in JC running comment | Text |
| HrRunning( ) | ^*slow* | Filter all starts with the word 'slow' found in JC running comment. This is a case-sensitive match. 'slow' will not match 'Slow' or 'SLOW'. | Text |

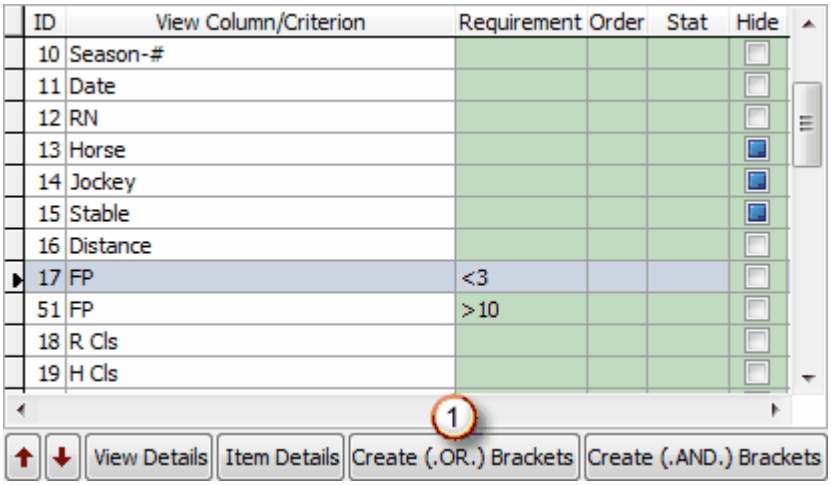
### 12.1.6 Creating and Removing OR or AND Criteria Group

#### Creating OR Criteria Group

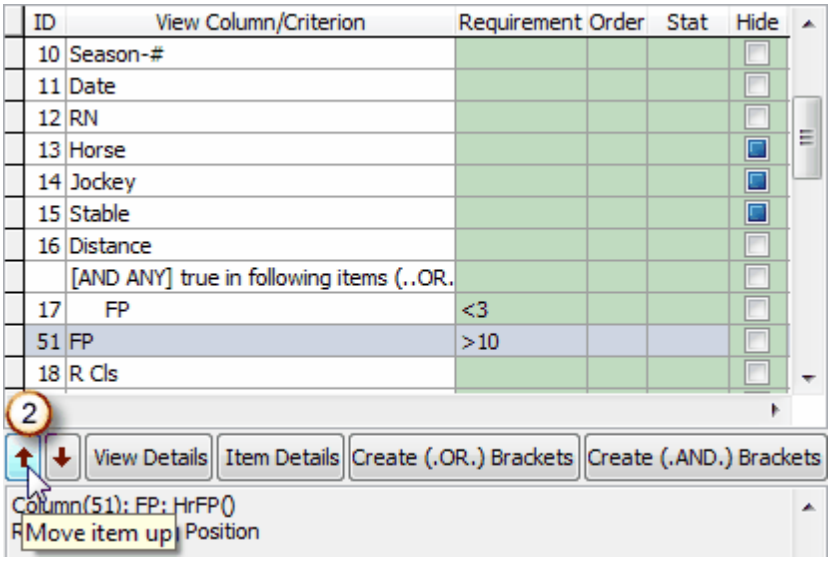
Follow the steps below to create an OR criteria group. The steps show how to create an OR group. Creating an AND group is similar.

1. Select a target requirement and click on the [Create (.OR.) Brackets] button. You will see a new OR group is created with the target item indented.

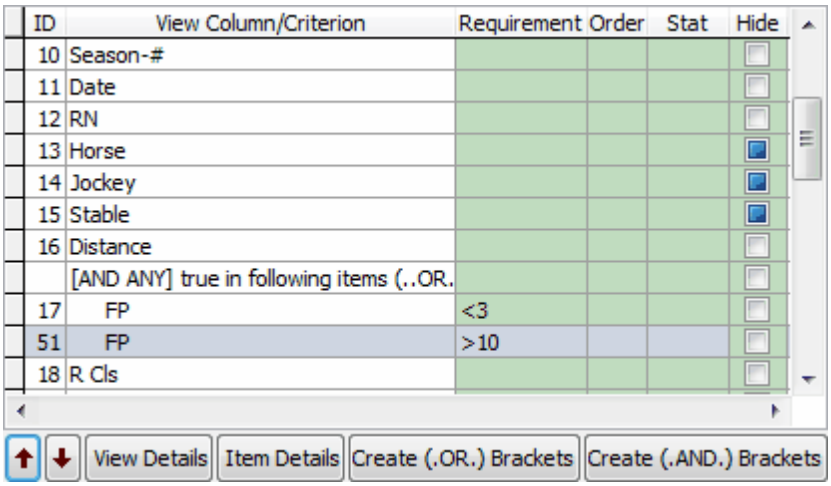
© Welkin Systems Limited




2. Select another target item. Click on the red brown UP or DOWN arrow buttons to move the item into the OR group. Repeat this step until all target items are moved into the group.




The picture below shows an OR group meaning "FP<3 OR FP>10" is created.




### Removing an OR or AND Criteria Group

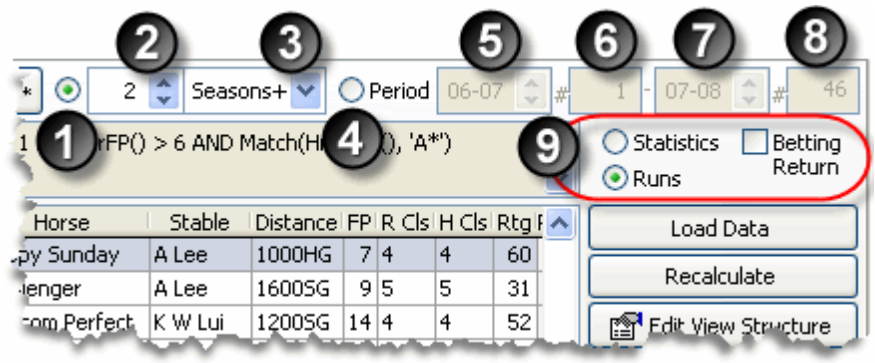
Use the red brown UP or DOWN arrow buttons to move all the group members out of

© Welkin Systems Limited

the group.

### 12.1.7 Data Period and Display Mode

The amount of data processed by a query can be set in the controls above the view data grid as shown in the figure below. When you set data periods in the controls, the data in the grid below will be shaded by grey lines showing that the data in the grid have not yet been refreshed to reflect the change in the controls. After a period is confirmed, click on the [Load Data] button to refresh the data and the grey lines will then disappear.




**Predefined Ranges**

When you click on the left option button (1), the following predefined ranges will be available in the range type dropdown list (3). You can set range by changing the value in the spinner (2) at the left of the dropdown. When you change the number, you will see the race numbers in the right boxes change accordingly to reflect the new setting.

| Range Type | Description |
|---|---|
| Seasons | Starting form the 1st race of a season and ending at the last races of the season. If you choose "1 season", only the races of the current season will be included. |
| Races | Number of races |
| Years | Number of calendar year |
| Months | Number of months |
| Days | Number of days |

There are 2 twin items for each range type, one with a "+" sign and the other without the sign. The ending date of all ranges will be the nearest racing date of the target (horse, jockey, etc). If we choose the one with a "+" sign, the nearest racing date means date of the coming run. If we choose the one without a "+" sign, the ending date will be the date of last run. Below is an example for setting range in the Horse form for the horse "Top Stitch".

Let us suppose:

1. Today is 10 Sep 2007
2. The horse "Top Stitch" will start on 12 Sep 2007
3. Latest start of "Top Stitch" was on 27 Jun 2007

© Welkin Systems Limited

If on the "Horse" form we set the data period to "1 Year":

All races during the period 28 Jun 2006 to 27 Jun 2007 will be included.

If on the "Horse" form we set the data period to "1 Year+":

All races during the period 13 Sep 2006 to 12 Sep 2007 will be included.

Under normal circumstances, if you are doing statistics, you should select the type without the "+" sign because results are not yet available for coming races, hence statistics can only be done for past races. If you are retrieving data for viewing only, you should select the type with a "+" sign, because you want to see information for the current (coming) meeting.

**Defining a Period by Race Numbers**

Instead of selecting a predefined range, you may click on the [Period] option button (4) to enable the race number range selection controls (5) – (8). You can then use the left season spinner (5) and the left race number textbox (6) to set a starting race, and the right season spinner (7) and the right race number textbox (8) to set an ending race. Unlike using pre-defined ranges, you can define a period which is far from recent meetings by defining race numbers.

**Data Display Modes**

There are two display modes (9) for displaying the data retrieved:

| Display Mode | Description |
|---|---|
| Runs | Show all records (runs) satisfying the data period and criteria set |
| Statistics | Group the records retrieved (i.e., what you see in the "Runs" mode) according to the grouping defined in the view structure and show the statistics you defined |

If you have not defined any groups in the view structure, all records will be regarded as a single group.

**Showing Betting Return**

If you check the [Betting Return] check box (9), the following betting return information will be shown:

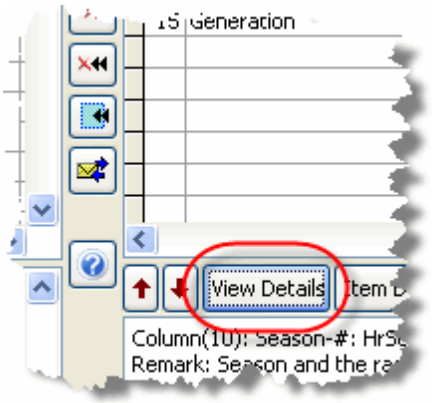
| Display Mode | Betting Return Information |
|---|---|
| Runs | Finishing position and Dividends |
| Statistics | • Start counts<br>• Wins counts<br>• WQP counts<br>• Lost counts |

© Welkin Systems Limited

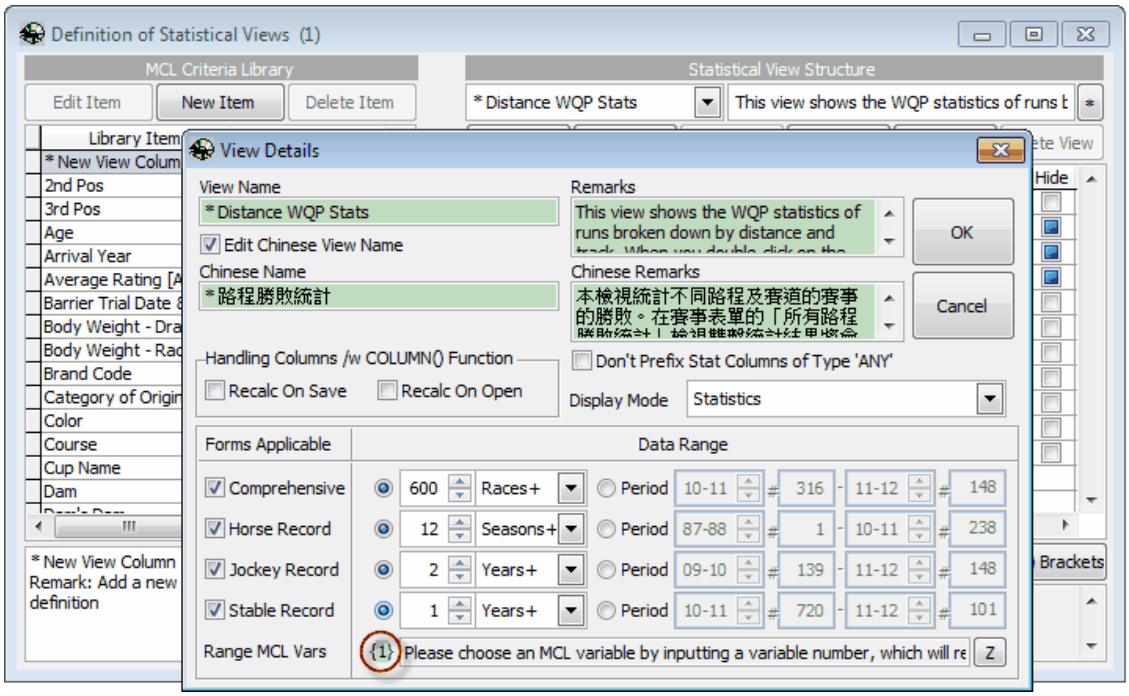
| | |
|---|---|
| | • Return on investment if betting on Wins (1.05 means 5% profit while 0.97 means 3% lost) |

### Setting Data Period in View Details form

Both data period and display mode can also be set on the "View Details" form, which can be opened by clicking on the [View Details] button on the "Definition of Statistical Views" form.




Once you uncheck any of the [Forms Applicable] checkboxes, the view will not be available in the corresponding form(s) and the data range controls will not be shown on the "View Details" form.



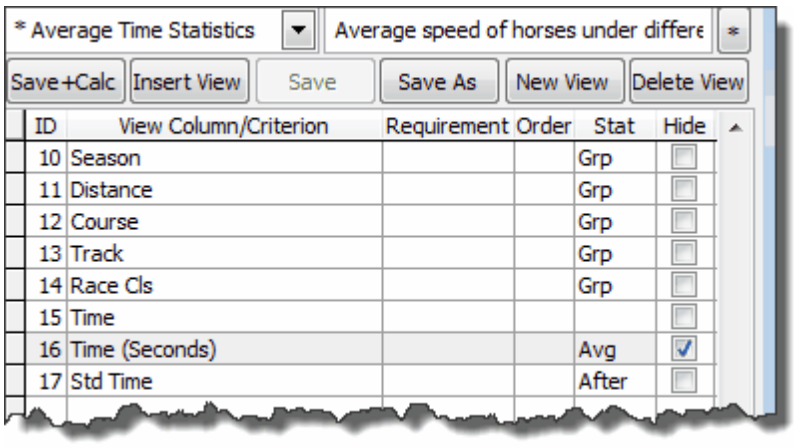

© Welkin Systems Limited

### 12.1.8 Statistical Items - the [Stat] Column

The "Stat" column in the "Definition of Statistical Views" form defines how data are grouped for statistics and what types of statistics on what columns are going to be carried out. The following table shows the available options.

| Option in Stat Column | Remark |
|---|---|
| Grp | Identical values will be regarded as a group. If no group is defined in a view, all records will be regarded as a single group. |
| Avg | Average in group (NULL values will not be counted) |
| Std | Standard deviation in group (NULL values will not be counted) |
| Var | Variance in group (NULL values will not be counted) |
| Sum | Sum in group |
| Cnt | Record count in group (NULL values will not be counted) |
| Max | Maximum in group |
| Min | Minimum in group |
| Any | An arbitrary value in group |
| After | The formula in this column will be evaluated after all statistics are done and before the results are shown. Usually this type of columns is for converting non-intuitive statistical results into readable or presentable values. |

Below is a screen capture of the structure of the "* Average Time Statistics" built-in view.




All races with the same season, distance, course, track and race class will be put into the same group. The average time value of each group is calculated and put into the [Std Time] column of the result table as shown below.

© Welkin Systems Limited

| Season | Distance | Course | Track | Race Cls | Std Time |
|---|---|---|---|---|---|
| 2005 | 1000 | HV | A | 4 | 0.58.1 |
| 2005 | 1000 | HV | B | 3 | 0.57.8 |
| 2005 | 1000 | HV | B | 4 | 0.58.5 |
| 2005 | 1000 | HV | B | 5 | 0.58.4 |
| 2005 | 1000 | HV | C | 2 | 0.57.9 |
| 2005 | 1000 | HV | C | 3 | 0.58.5 |
| 2005 | 1000 | HV | C | 4 | 0.58.5 |
| 2005 | 1000 | HV | C | 5 | 0.58.6 |
| 2005 | 1000 | HV | C+3 | 5 | 0.58.6 |
| 2005 | 1000 | SH | A | 3 | 0.58.1 |
| 2005 | 1000 | SH | A | 4 | 0.58.5 |

Please note the difference among of columns 15, 16 and 17 :

- Column 15 returns time values in a string format similar to "0.57.4". Unlike numbers, character strings can't be used for calculating averages. Therefore this column is only for displaying time values in the Runs mode and will not participate in any statistics. Even if this column is removed, the statistical results will not be affected.
- Column 16 returns time values in numbers of seconds, which will be used for the "average" calculation. However most users are not used to the "seconds" format of time. This column is marked "Hide" and will neither be shown in Runs mode nor Statistics mode.
- Column 17 will be evaluated after all statistics are done and before results are shown. This column converts time values from the "seconds" format to a readable "string" format by using the SToT( ) function and will be used for displaying the statistical result.

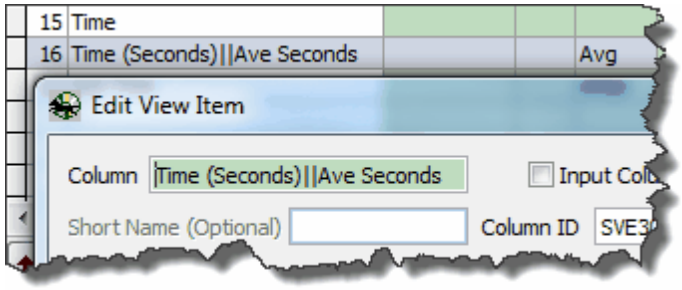
**Heading of a Statistical Column**

1. By default the heading of a statistical column (not including Grp and After) in Statistics mode will be identical to the heading of the column in Runs mode except that a "type label" prefix will be added. For example, if we clear the [Hide] checkbox of column 16 to make it show in the statistical result table, its column heading will be "avg:Time" as shown below.

| Season | Distance | Course | Track | Race Cls | avg:Time | Std Time |
|---|---|---|---|---|---|---|
| 2006 | 1400 | SH | B | 2 | 86.58 | 1.26.6 |
| 2006 | 1400 | SH | B | 3 | 83.19 | 1.23.2 |

2. As stated in the above point, the label "any:" will normally appear in the heading of columns of "Any" type. However, you can suppress it by clearing the [Don't Prefix Stat Columns of Type 'ANY'] checkbox in the "View Details" form (shown by clicking on the [View Details] button on the "Definition of Statistical Views" form.
3. If you don't like the default heading of a statistical column of the statistical results, you can assign a specific heading to it by putting this heading after the normal heading and separate them by a "|" character. The space, if any, surrounding the "|" character will not be shown. For the column as shown in the figure below, the column heading will be "Time" in the Runs mode while "ATime" in the Statistics mode.

© Welkin Systems Limited




### 12.1.9 Viewing Past Records and Move Columns in Stat View Page

Show Records of Horse, Jockey and Stable

To view past records of Horse, Jockey and Stable, please follow the steps below. The step numbers are circled in red on the screen captures.

1. Open the data form and switch to the [Stat View] page.

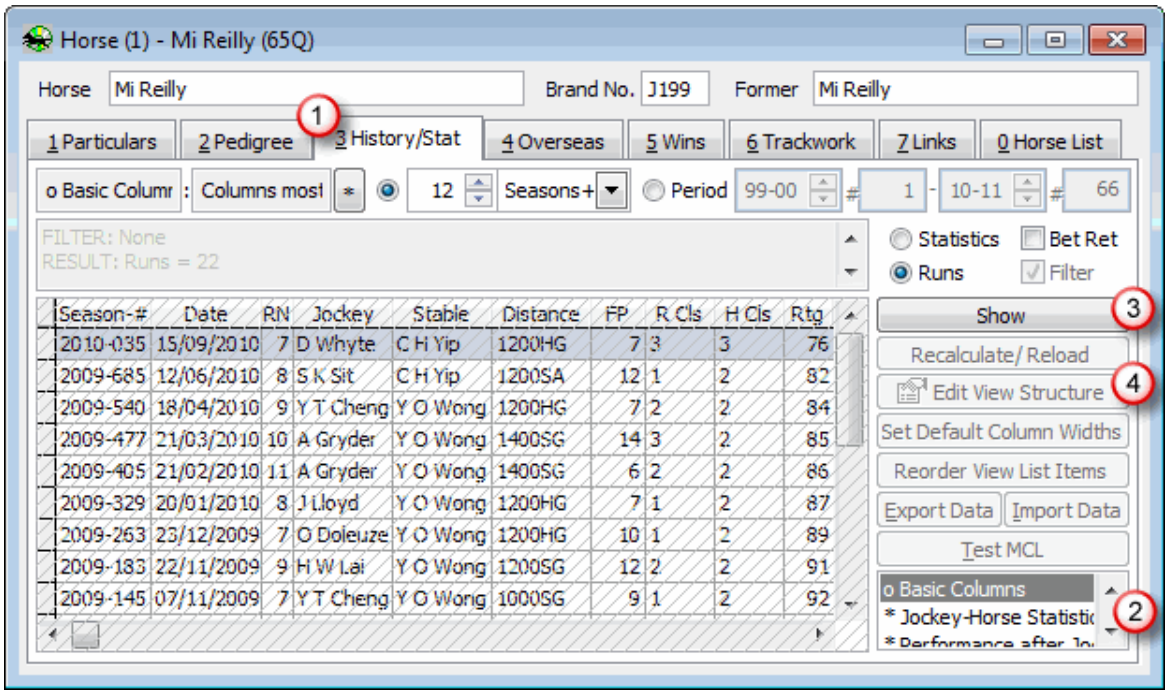
At the right bottom of the page, you will find a view list.

2. Click on the [o Basic Columns] view name.

If you have changed from another view to the [o Basic Columns] view, you may see that the data area is shaded by grey lines. (If there was no view change, grey lines will not show.)

3. Click on the [Load Data] button.

Data will be loaded and the grey shading, if any, in the data area will disappear.




© Welkin Systems Limited

### Rearranging Column Positions

After loading the data, you may want to move the columns that you frequently use to the leftmost of the data grid. Let us try moving the "FP" column up so that it just follows the "Horse" column.

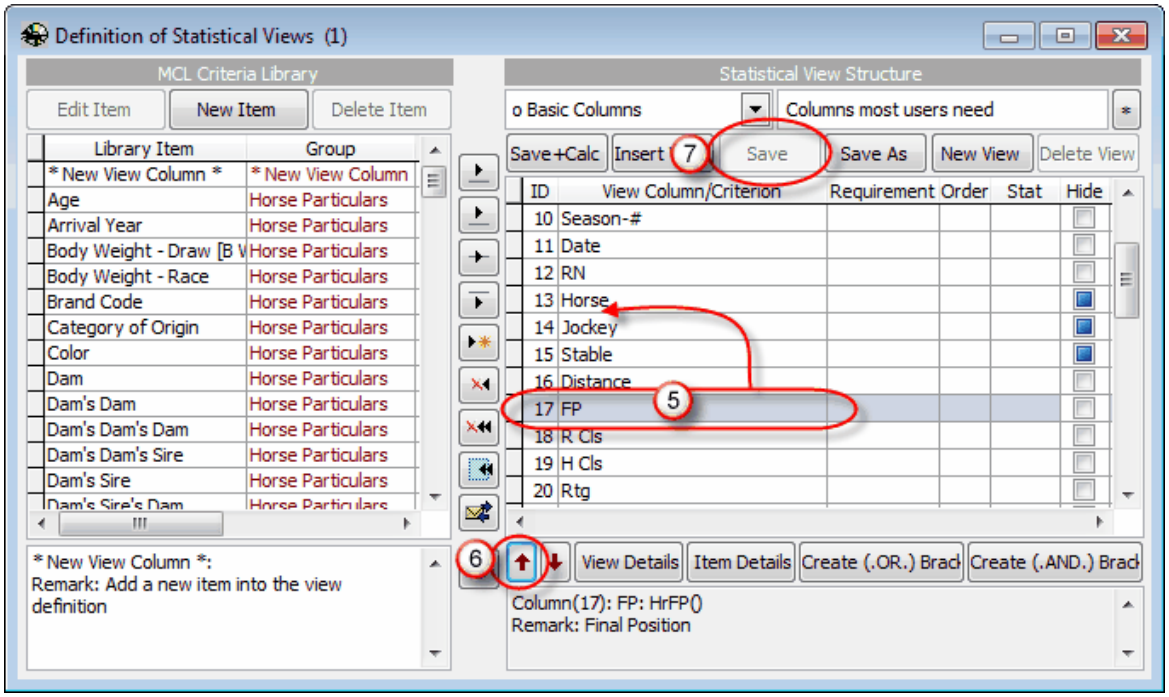
4. Click on the [Edit View Structure] button on the Horse form

The "Definition of Statistical Views" form will be opened showing the structure of the [o Basic Columns] view.

5. Click to highlight the "FP" column

6. Click on the [Up] key 3 times at the right bottom of the grid

The "FP" column will be move up to the location just below the "Horse" column.




7. Click on the [Save] button

Close the "Definition of Statistical Views" form. Go back to the Horse form to see the change. Please make sure the [o Basic Columns] view is selected.

### Moving Other Columns

In order to move other columns, click again on [Edit View Structure] button of the Horse form to open the "Definition of Statistical Views" form. You may move more than one column into new positions before saving your change.

© Welkin Systems Limited

## 12.2 Type 2: Statistics via Starter Views (Race Data Form)

Statistical View is a sophisticated tool for statistics in RaceMate. A statistical view is composed of MCL columns, which retrieve information from the database of RaceMate. Grouping and actual statistics are defined by assigning functions to view columns. What a statistical view can do is only limited by your imagination.

We will use a simple example to explain how to carry out statistics with a statistical view.

For more examples please see:
Find horses which win in 60 days after a stable change

See the Type 1: Starter Views section for more information about starter views.

**>>> Subtopic of this section <<<**

- Betting Index
- Evaluating Accuracy of Prediction Models
- Example: Draw Position Statistics

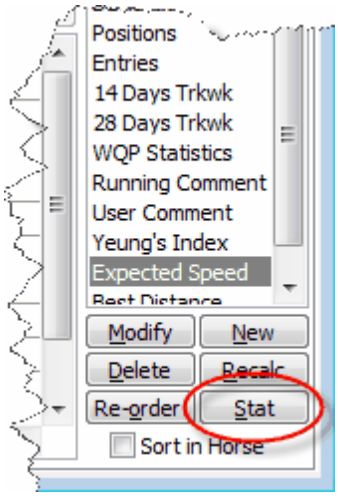
### 12.2.1 Betting Index

Many of the starter views are actually prediction models which help you betting. If you want to evaluate the accuracy of your prediction models, you must create a column and mark it as "betting index". When populating figures into the betting index column, you must give the horse with higher chance to win should have a higher value.

### 12.2.2 Evaluating Accuracy of Prediction Models

Normally a view which acts as a prediction model should have a betting index column. Follow the steps below to evaluate the accuracy of the betting index.

1. Click on the [Stat] button under the view list.

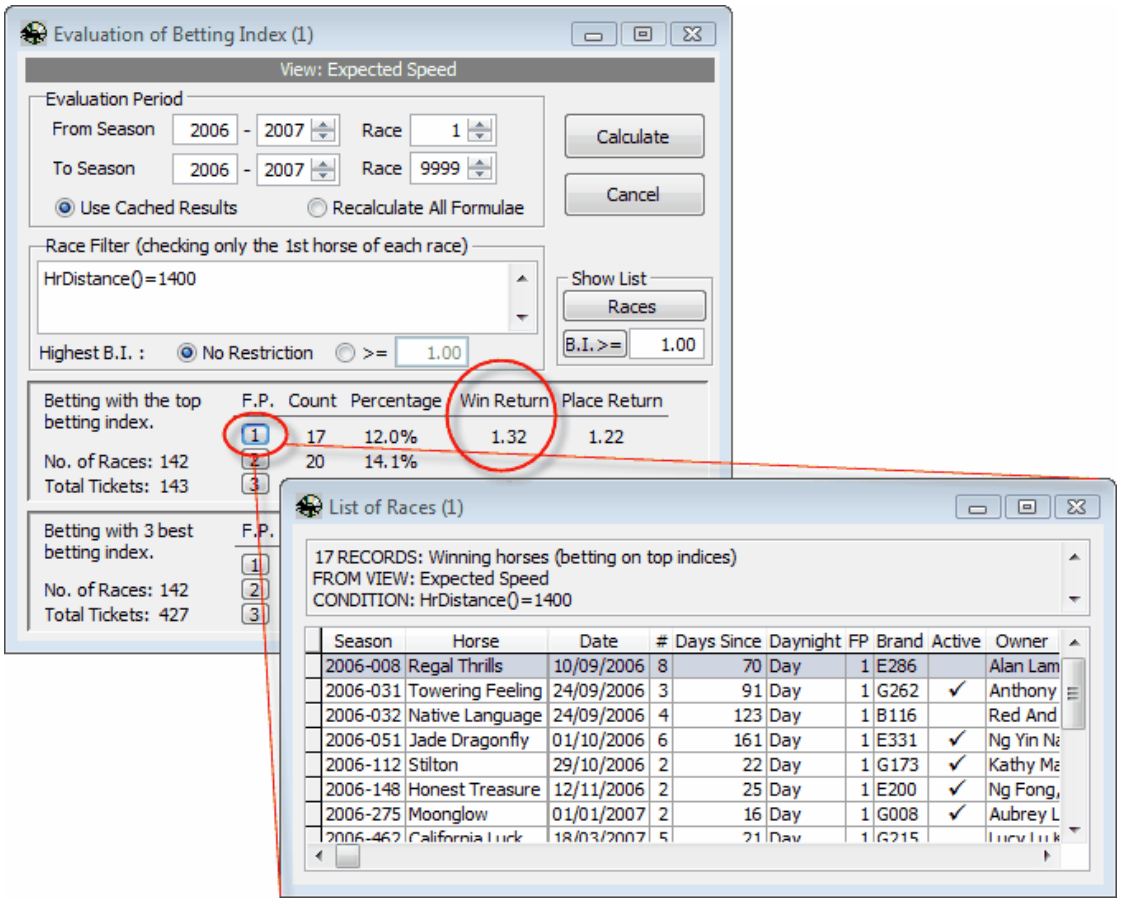



2. When the "Evaluation of Betting Index" form (figure below) shows, you may do a number of things.
   - You may set a evaluation period

© Welkin Systems Limited

- Every time you open the a view, the cell values of the view will be cached into the Cache folder. If you are confident of these cached results, you may choose [Use Cached Results]. On the other hand if you want to recalculate all cell values, choose [Recalculate All Formulae].
- You may optionally define a filter to carry out the evaluation only on the races satisfying your filter
- You may exclude some races by setting the [Highest B.I.] restriction

3. Click on the [Calculate] button to start the evaluation. Results will be shown on the lower part of the form. On the form below, you can see that if you have betted on Wins only (according to the betting index of the "Expected Speed" view), you should have won 32% (1 dollar betting returns 1.32 dollars)


Evaluation of Betting Index (1)
View: Expected Speed
Evaluation Period
From Season 2006 - 2007 Race 1
To Season 2006 - 2007 Race 9999
Use Cached Results
Recalculate All Formulae
Calculate
Cancel
Race Filter (checking only the 1st horse of each race)
HrDistance()=1400
Highest B.I. : No Restriction >= 1.00
Show List
Races
B.I.>= 1.00
Betting with the top betting index.
No. of Races: 142
Total Tickets: 143
F.P. Count Percentage Win Return Place Return
1 17 12.0% 1.32 1.22
2 20 14.1%
3
Betting with 3 best betting index.
No. of Races: 142
Total Tickets: 427
List of Races (1)
17 RECORDS: Winning horses (betting on top indices)
FROM VIEW: Expected Speed
CONDITION: HrDistance()=1400
Season Horse Date # Days Since Daynight FP Brand Active Owner
2006-008 Regal Thrills 10/09/2006 8 70 Day 1 E286 Alan Lam
2006-031 Towering Feeling 24/09/2006 3 91 Day 1 G262 ✓ Anthony
2006-032 Native Language 24/09/2006 4 123 Day 1 B116 Red And
2006-051 Jade Dragonfly 01/10/2006 6 161 Day 1 E331 ✓ Ng Yin Na
2006-112 Stilton 29/10/2006 2 22 Day 1 G173 ✓ Kathy Ma
2006-148 Honest Treasure 12/11/2006 2 25 Day 1 E200 ✓ Ng Fong,
2006-275 Moonglow 01/01/2007 2 16 Day 1 G008 ✓ Aubrey L

4. If you want to view all the winning races, click on the [1] button to show the list.

## 12.2.3 Example: Draw Position Statistics

You may be one of the people are interested in how draw positions affect the performance of horses. We are going to create a simple statistical model for finding out their relations.

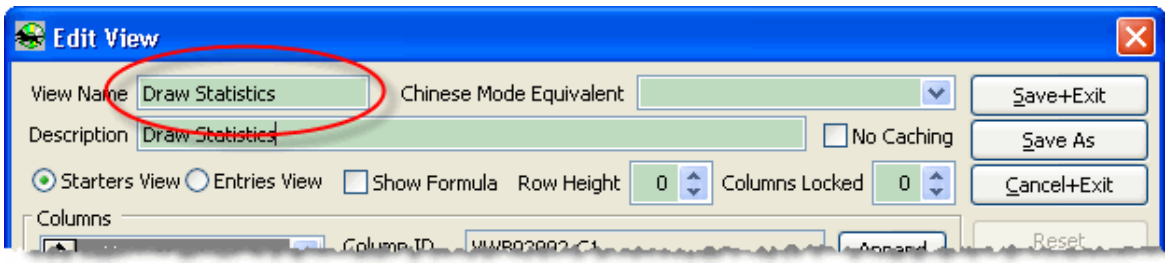
**>>> Subtopics of this section <<<**

© Welkin Systems Limited

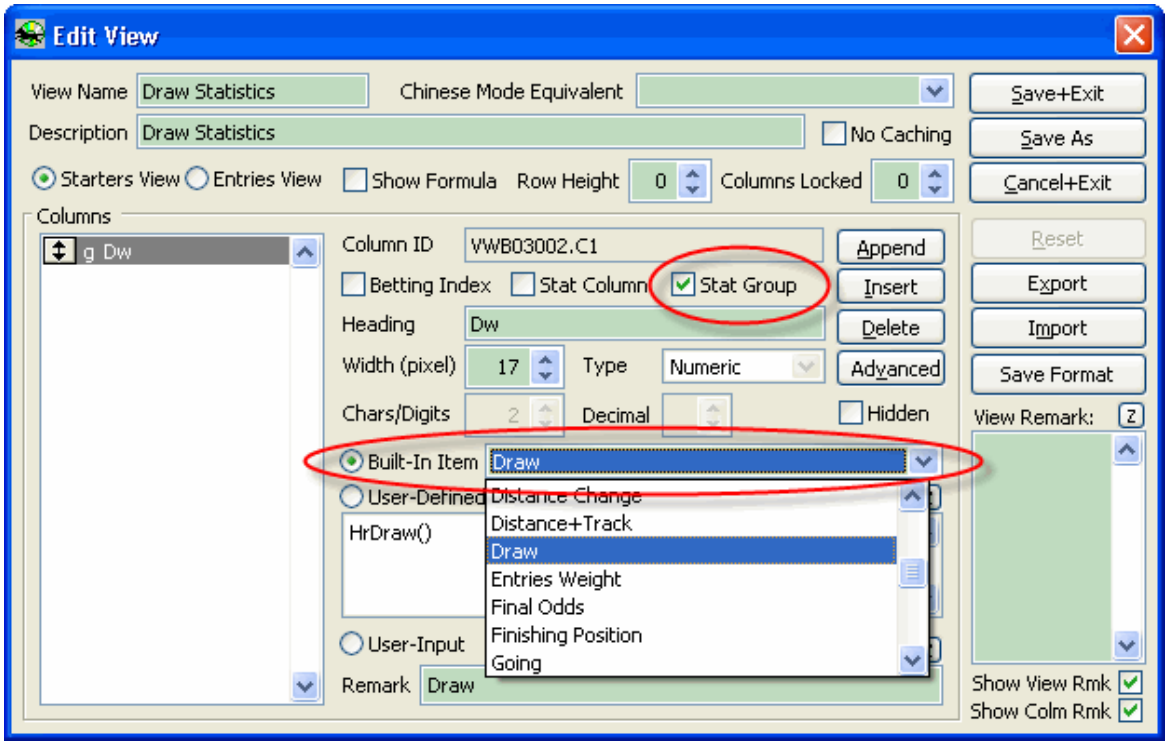
- Basic Steps
- Statistical Groups
- Printing or Exporting Statistics

#### 12.2.3.1 Basic Steps

1. Open the Race Data form
2. On the Starters View page, click on the [New] button to create a new view
3. Input "Draw Statistics" into the View Name box

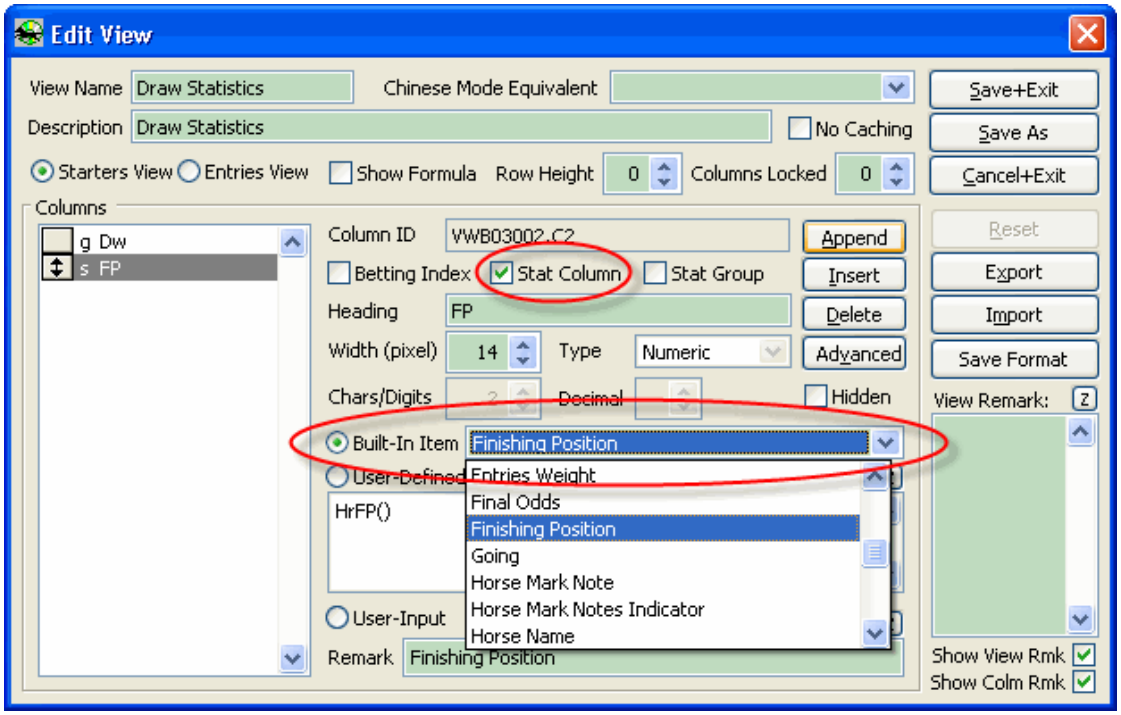



4. Delete the default column 'Horse'. Add a new built-in column 'Draw' into the view and set it as [Stat Group].

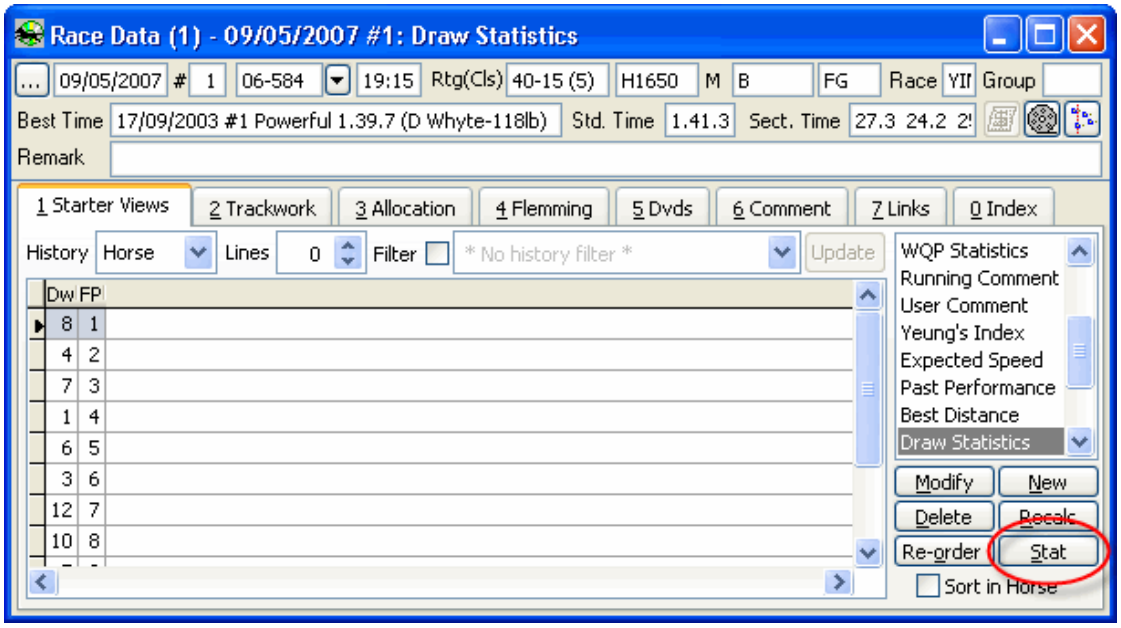



5. Add another new built-in column "Finishing Position" and set it as [Stat Column]

© Welkin Systems Limited

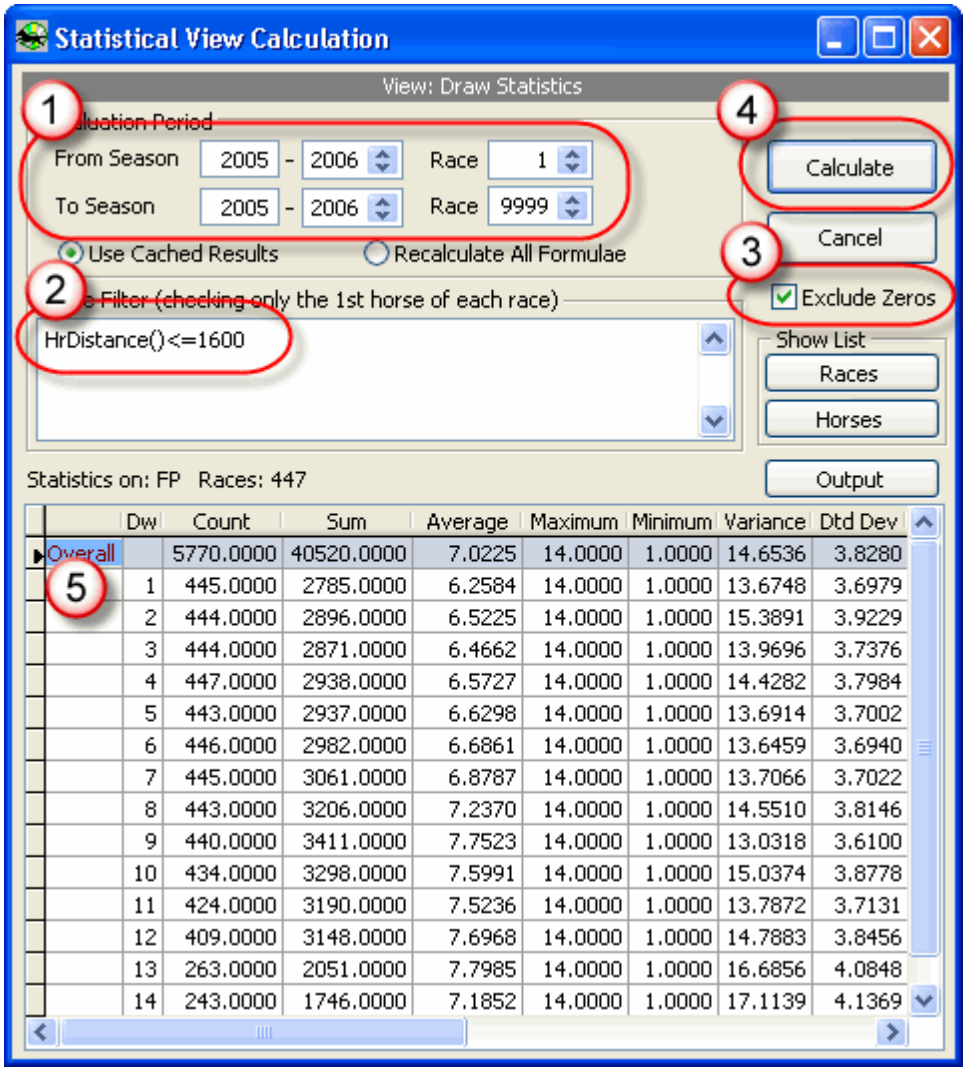



6. Click on the [Save+Exit] button to save and exit the Edit View form. You will see a view page similar to the following screen.
7. Click on the [Stat] button.




8. You will see the following Statistical View Calculation form
9. Set the period to "from the 1st race of season 2005-2006 to the last race (9999) of the season 2005-2006".
10. Enter the following expression into the Criteria Filter box:
    HrDistance( )<=1600
11. Make sure [Exclude Zeros] is checked

© Welkin Systems Limited



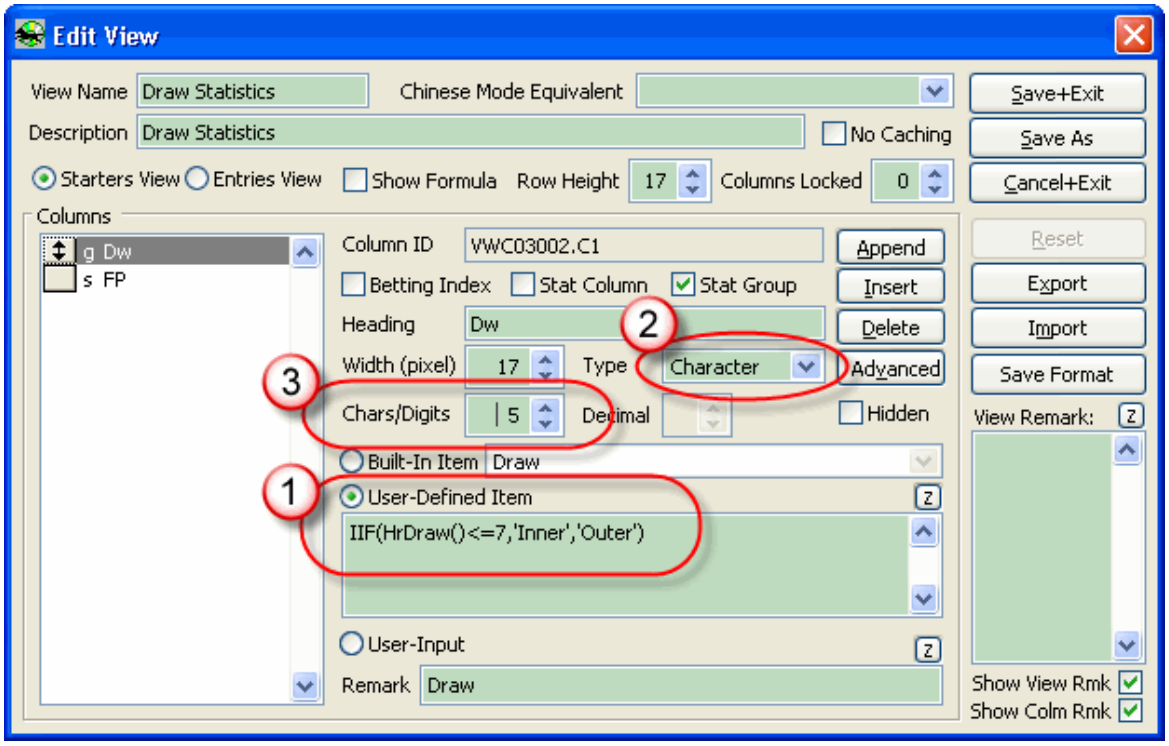

12. Click on the [Calculate] button. You will then see the statistical result shown in the lower part of the form. The result shows that the average finishing position of all draw positions is 7.0225. Position 1 is best draw position, the average finishing position of which is 6.6514.

#### 12.2.3.2 Statistical Groups

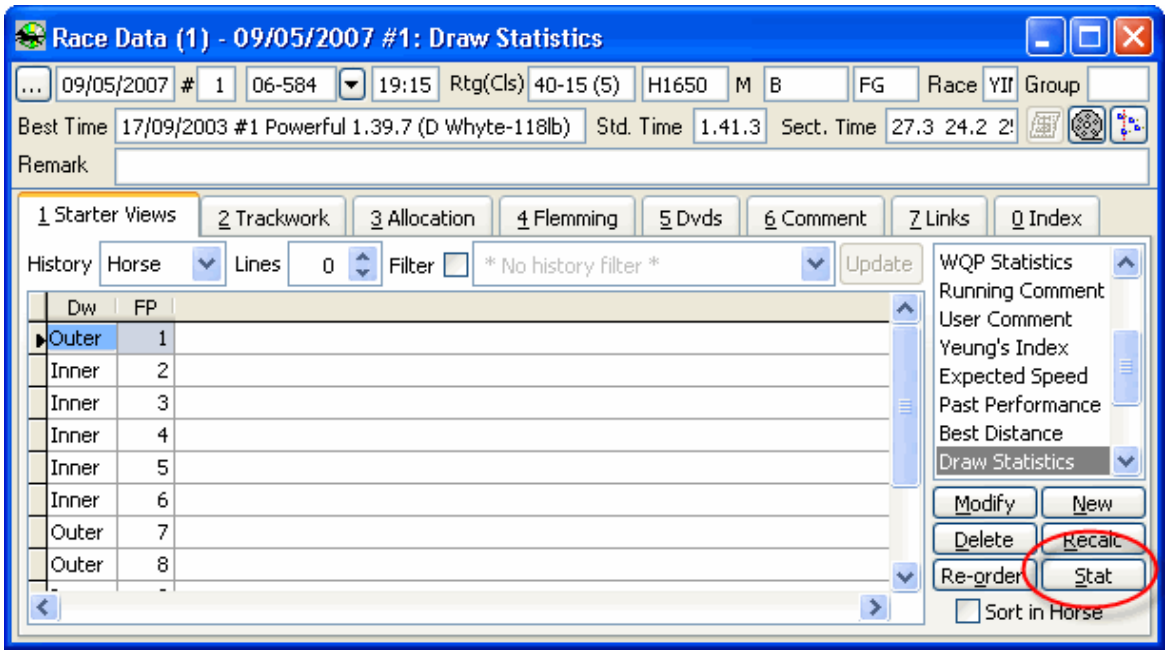
In order to compare the performance of the horses in inner and outer tracks, let us modify the "Stat Group" column of the statistical view of the above section.

1. Select the Draw Statistics view. Open the Edit View form again and select the Draw column.

© Welkin Systems Limited

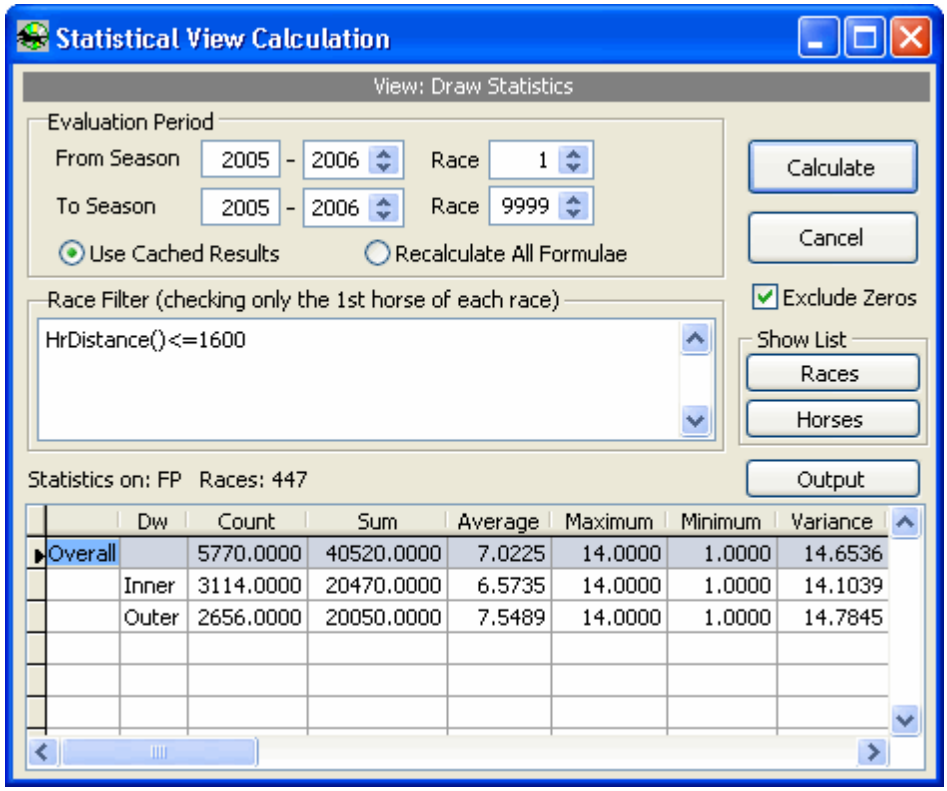



2. Change the column from [Built-In Item] to [User-Defined Item]
3. Change the [Type] of the column from Numeric to Character
4. Set the column width to 5




5. Click on the [Stat] button to show the following "Statistical View Calculation" form

© Welkin Systems Limited



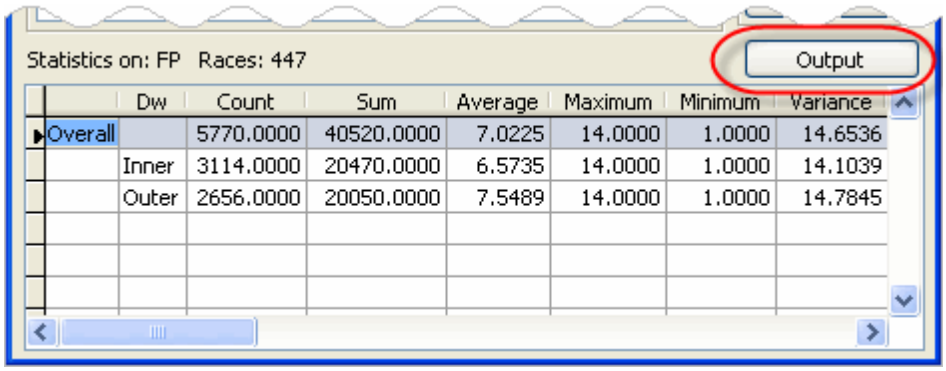

6. Keep the existing filter expression (HrDistance( )<=1600) and click on the [Calculate] button to calculate the result in the lower part of the form. The result shows that the average finishing position of Inner tracks is 6.5735, which is significantly better than that (7.5489) of the Outer tracks.

For more examples please see:

Find horses which win in 60 days after a stablechange

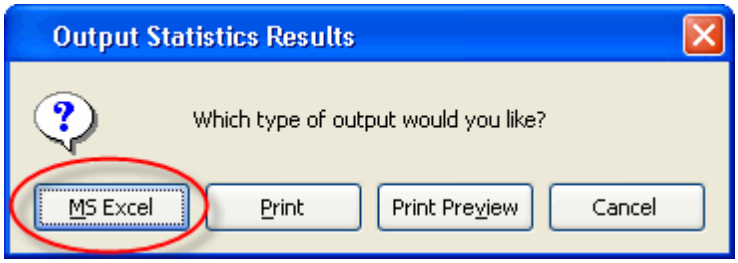
### 12.2.3.3 Printing or Exporting Statistics

You can output your statistics by clicking on the [Output] button of the form.



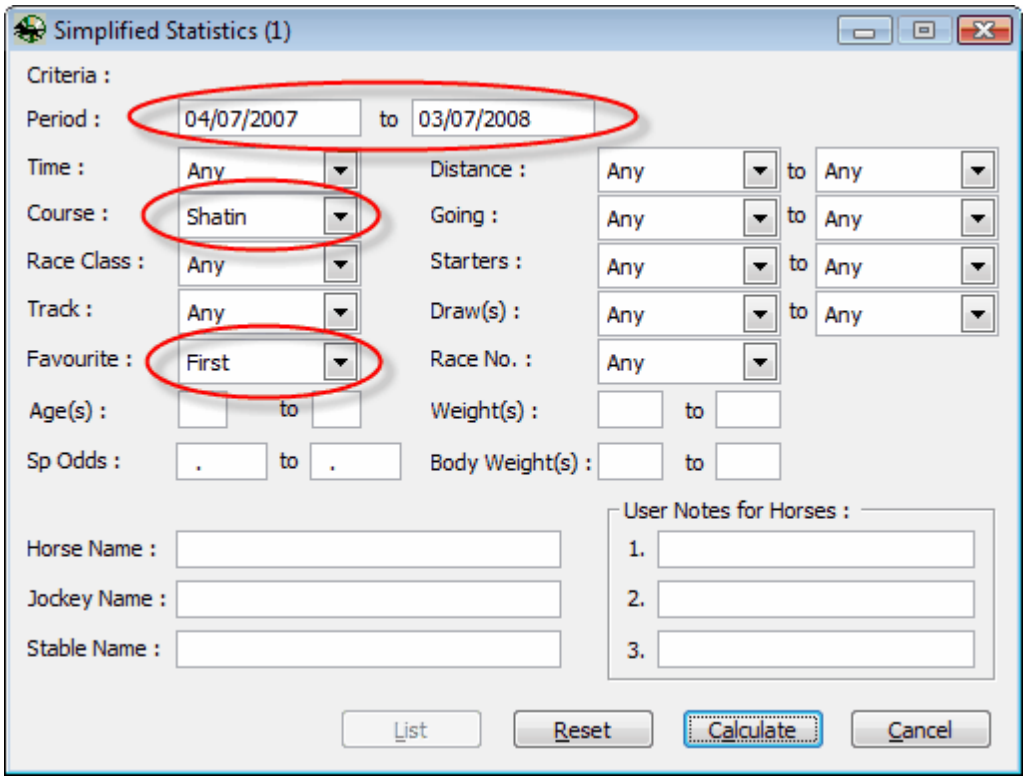

You will see the following dialog box. Click on the [MS Excel] button to export the result to MS Excel or [Print] button to output it to the printer.

© Welkin Systems Limited



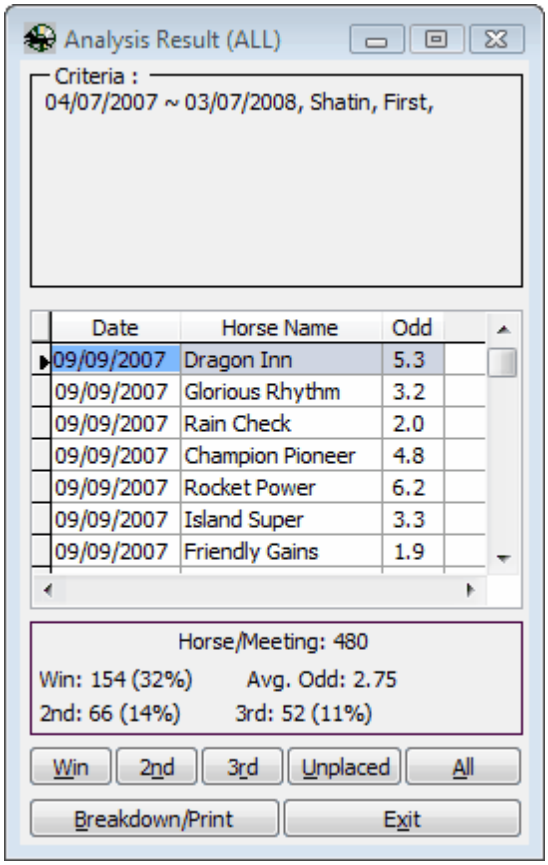

## 12.3 Type 3: Simplified Statistics Form

Select [Simplified Statistics] from the [Analysis] menu to open the Simplified Statistics form. Then set the criteria you want on the form and click on the [Calculate] button.




The following form show the results.

© Welkin Systems Limited




The above form shows that if we have betted $10 in each race during the season 07-08, the total investment would then be $4,800. We would have won 154 races. Since the average figure of the odds was 2.75, the return would then be $4,235 and a lost $565 (-13%) would have been got.

| | | | | | |
|---|---|---|---|---|---|
| Total Investment | : | 480 x $10 | = | $ 4,800.00 | |
| Return | : | 154 x 2.75 x $10 | = | $ 4,235.00 | |
| | | | | $ -565.0 | (-13%) |

© Welkin Systems Limited

# Part

# 13 Exporting and Importing Data

This chapter discusses how to export data into external files. Microsoft Excel is the most common file format.

**>>> Subtopics of this section <<<**

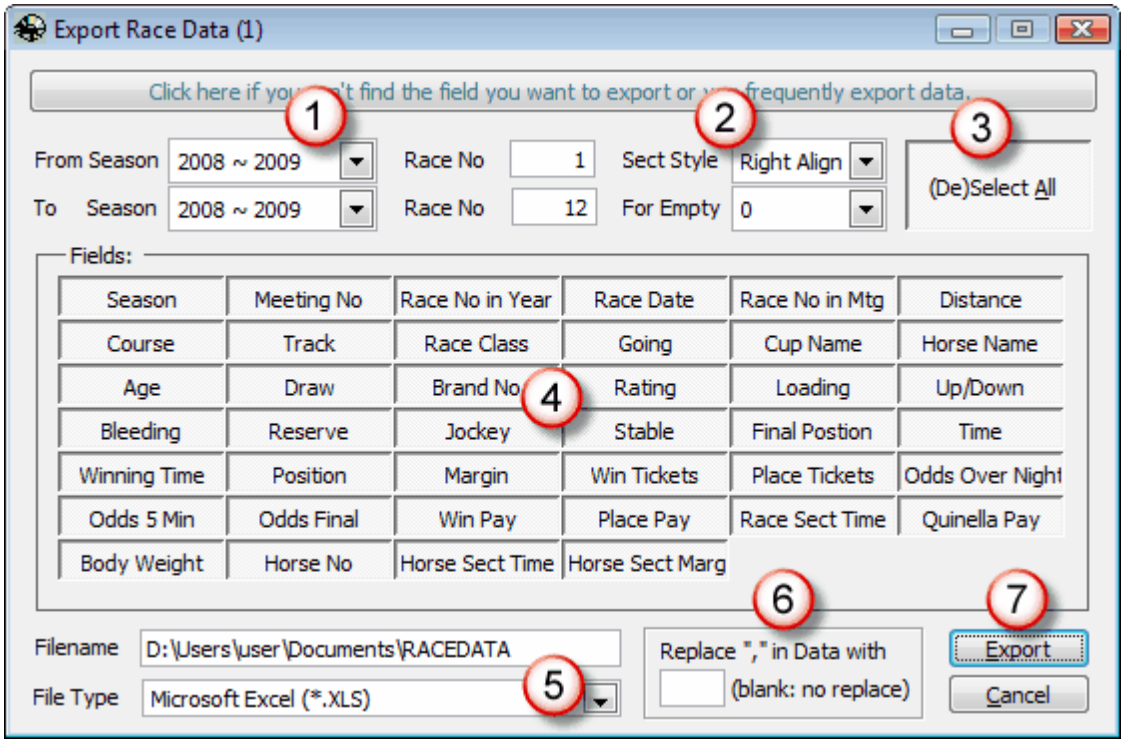
- Exporting Simple Race Data
- Exporting Horse Data
- Exporting or Copying Grids Cells
- Exporting View Data
- Data Export via Starter Views
- Format of Exported Excel Files
- Importing View Data From Excel Files

## 13.1 Exporting Simple Race Data

If you want to use other computer software (e.g., Microsoft Excel) to analyse race data, you could export data from RaceMate and then import the data from the exported file in to the target software.

The [Export Race Data] form discussed here includes only basic fields. If you want to export more complicated fields, you may choose [Tools] - [Comprehensive Statistics and Records Query] and then click on the [Edit View Structure] button to create a view with all the information you want and follow the steps in the Exporting View Data page to export the view to an Excel or text file. In the view, you may include RaceMate both built-in columns and tailor-made MCL columns.

Here are the steps for exporting basic race data. Choose [Export Race Data] from the [Tool] menu. You will see the following form.




© Welkin Systems Limited

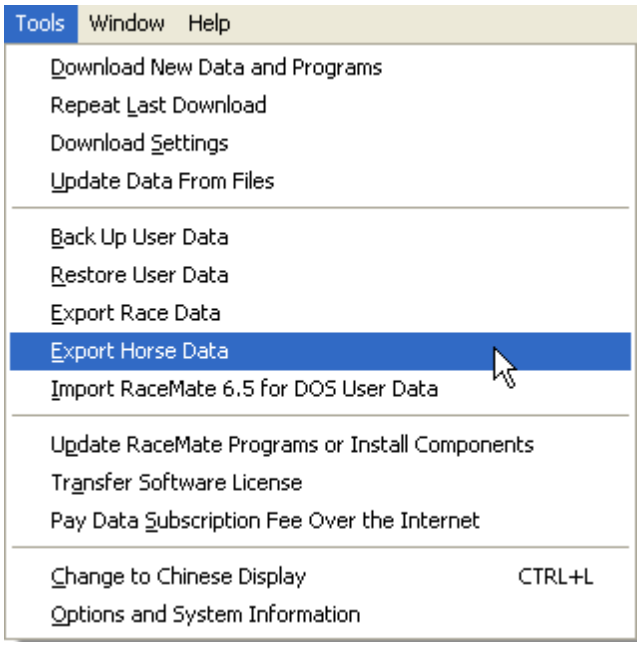
Steps:

1. Specify the period of interest.
2. The data to be exported may includes sectional data. For example you may want to export the section positions of a 4-section race. If you may choose [Left-Align], you will put the 4 positions of each horse into columns 1, 2, 3, 4 and leave columns 5 and 6 blank (there are 6 columns for positions). If you choose [Right-Align], you will put the positions into columns 3, 4, 5, 6 and leave columns 1, 2 blank. Please choose the way of representation for empty cells in the "For Empty" drop-down.
3. Click [(De)Select All] to select all fields button. If you click once again the same button, all fields will be deselected.
4. If necessary, fine-tune your selection by choosing proper buttons in the "Fields" area.
5. Enter a file name and choose a file format.
6. If you are going to export the data into a text file, commas are the field separators. If there is any "," inside the data to be exported, fields may be misplaced. You may enter a symbol to replace "," in data.
7. Click on the Export button to export the data into the file

## 13.2 Exporting Horse Data

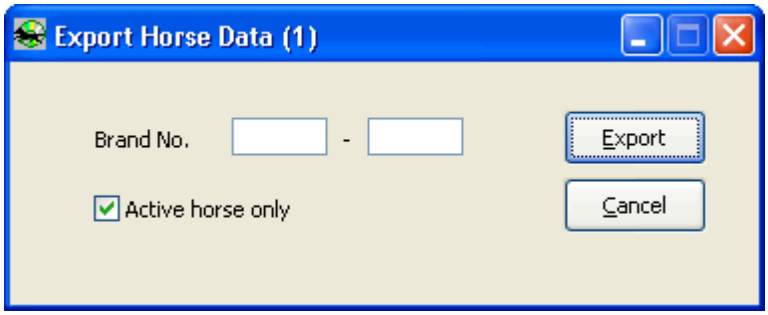
Follow the steps below to export horse data into a file.

1. Choose [Export Horse Data] from the [Tools] menu




2. When the Export Horse Data form shows, enter the range of Brand Numbers into the text boxes of the form

© Welkin Systems Limited

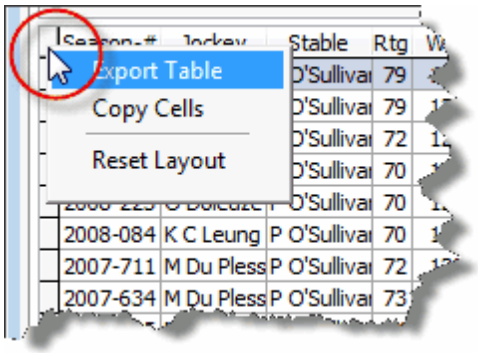



3. Click on the [Export] button to export the data into an Microsoft Excel file.

## 13.3 Exporting or Copying Grids Cells

### Exporting Data to Excel Files

In RaceMate all data grids are exportable to Excel. You may right click on the left top corner of a grid on any data form and choose [Export Table] to export the content of the grid to an Excel file or a text file.




### Copying All Cells to Clipboard

You may choose [Copy Cells] of the above menu to copy the whole table into the Windows clipboard. The data copied can be pasted into other applications.

### Advance Data Export

You may create a view containing all the data you want to export. Click on the left-top corner of the grid to export all the data in the view. Besides built-in columns you may create complicated columns using MCL functions for the export. Both "Comprehensive Statistics and Record Query" (as well as the History/Stat page of Horse, Jockey, Stable forms) or "Starter" views for such export action. Generally speaking the former one may be simpler for most cases.

## 13.4 Exporting View Data

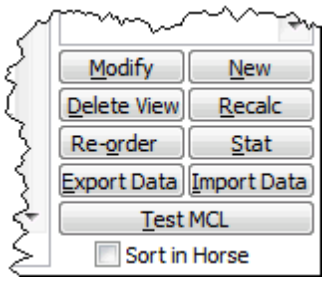
You can use the [Export Data] button on a view page to export view data into Excel files. These Excel files is actually tab-delimited text files with and "xls" extension rather than true Excel files. (See Format of Exported Excel Files to learn more about this file type.)

© Welkin Systems Limited

On the "Export Race Data" form, you can export only some basic data. If you want to export extensive data, including statistics, you should create a view with all items you want (using proper MCL functions). Then export the view.

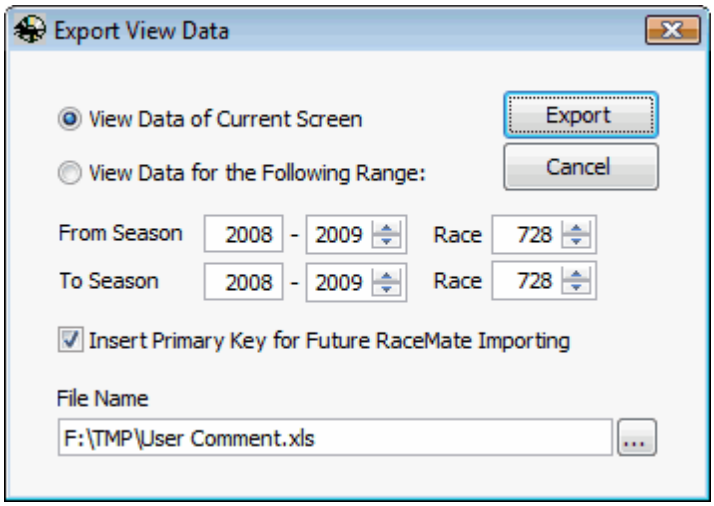
### Exporting Data of Starter Views

Click on the [Export Data] button on the "Starter Views" page.




You will then see the following "Export View Data" form. The following options are shown on the form.

1. Range of data to be exported
2. Whether the exported file will be imported. If yes, a primary key will be added at the left. Please see Importing View Data From Excel Files to learn more about the primary key.
3. Name of the exported file

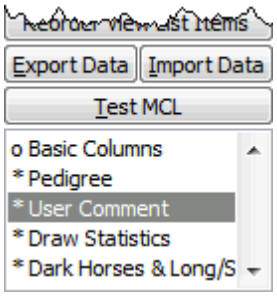

Export View Data
View Data of Current Screen
View Data for the Following Range:
Export
Cancel
From Season 2008 - 2009 Race 728
To Season 2008 - 2009 Race 728
Insert Primary Key for Future RaceMate Importing
File Name
F:\TMP\User Comment.xls

Click on the [Export] button start exporting.

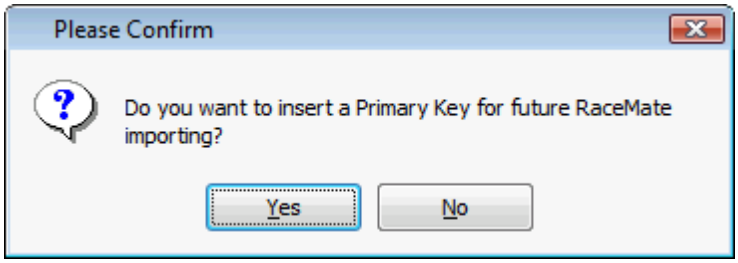
### Exporting Data of History/Stat Views

Click on the [Export Data] button on the "History/Stat Views" page.

© Welkin Systems Limited

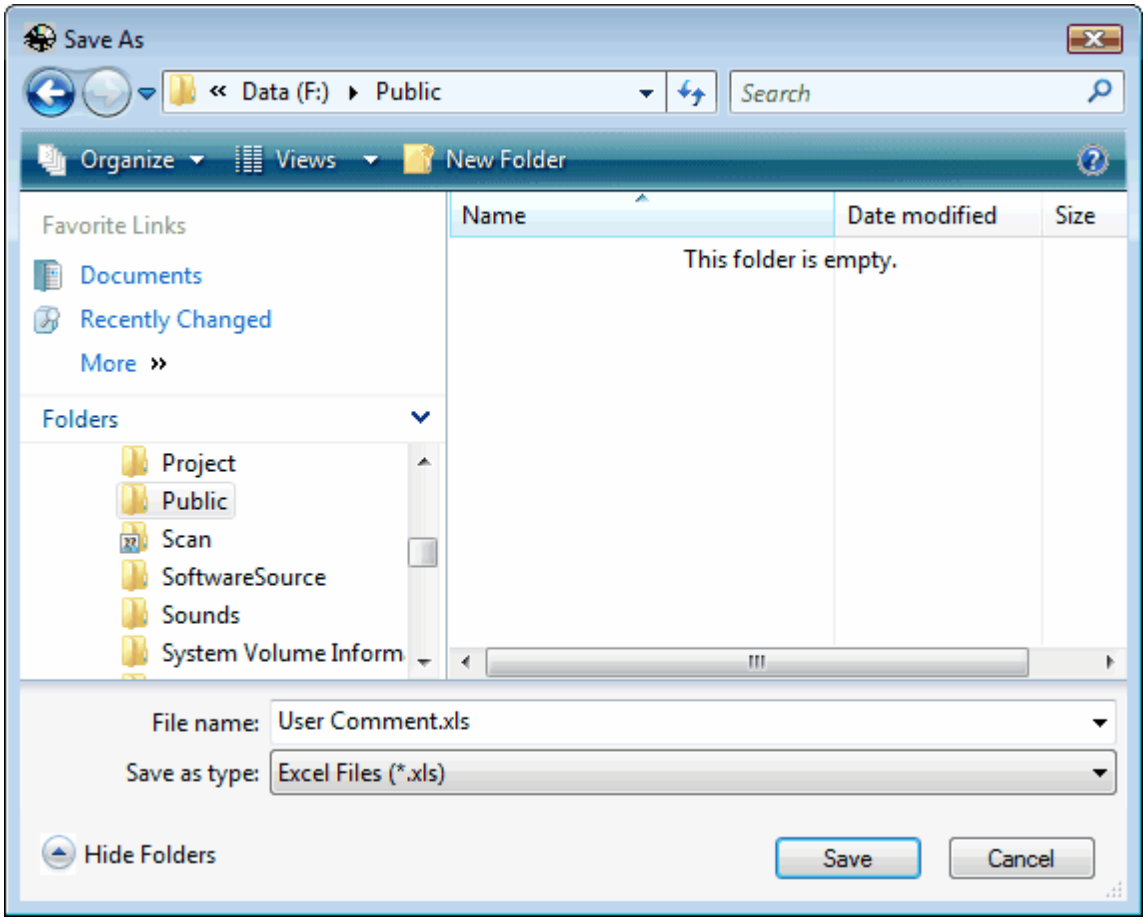



If you are showing details of races and there is at least a input column on the view, you will see the following question. Otherwise you will go to the "Save As" dialogue directly.




If you answer [Yes] on the above dialogue box, a primary key column will be created as the first column of the exported file. (See Importing View Data From Excel Files to learn more about the primary key.) No matter what answer you choose, you will see the following "Save As" dialog. The view name will be the default file name.

© Welkin Systems Limited




Click on [Save] to export the view.

## 13.5 Data Export via Starter Views

Under normal circumstances, the technique described in Exporting Contents of Grids in Forms will be sufficient for all complex data export. Use the technique of this page only when necessary.

Combining the flexibility of views and the grouping feature of view statistics, you will have the absolute power to retrieve the data of interest into your data export files. It is powerful because:

- What we can do in other export functions, including Exporting Race Data, and Exporting Horse Data, discussed in previous sections, can all be done here.
- By using Data Export via Starter Views, you can export not only the normal race and horse data, but also your notes, ratings, statistical results, jockey information, stable information, and much more.
- Full power of MCL is available in Data Export via Starter Views.

Example

We will demonstrate the basic steps of running Data Export via Starter Views by a practical example. In the example, we will use the "User Comments" view to export the

© Welkin Systems Limited

data columns in the following list. Although "User Comments", which we use in this example, is a built-in view, in most cases we will create a new view for Data Export via Starter Views.

| Existence | Column to Export | Type | MCL Function |
|---|---|---|---|
| Built-in | Horse Name | Existing built-in column | |
| | No. | Existing built-in column | |
| | Rtg | Existing built-in column | |
| | R1<br>(User rating) | Existing built-in column<br>Numeric(3,0) | |
| | Comment<br>(User comment) | Existing built-in column<br>Character(200) | |
| New | Season-# | New built-in column | |
| | FP<br>(Finishing position) | New built-in column | |
| | Top3 Rtg<br>(Rating of 3 fastest horses) | New user-defined column<br>Character(12) | RunnersList('HrRating( )',,'For HrFp( )<=3') |

**Steps**

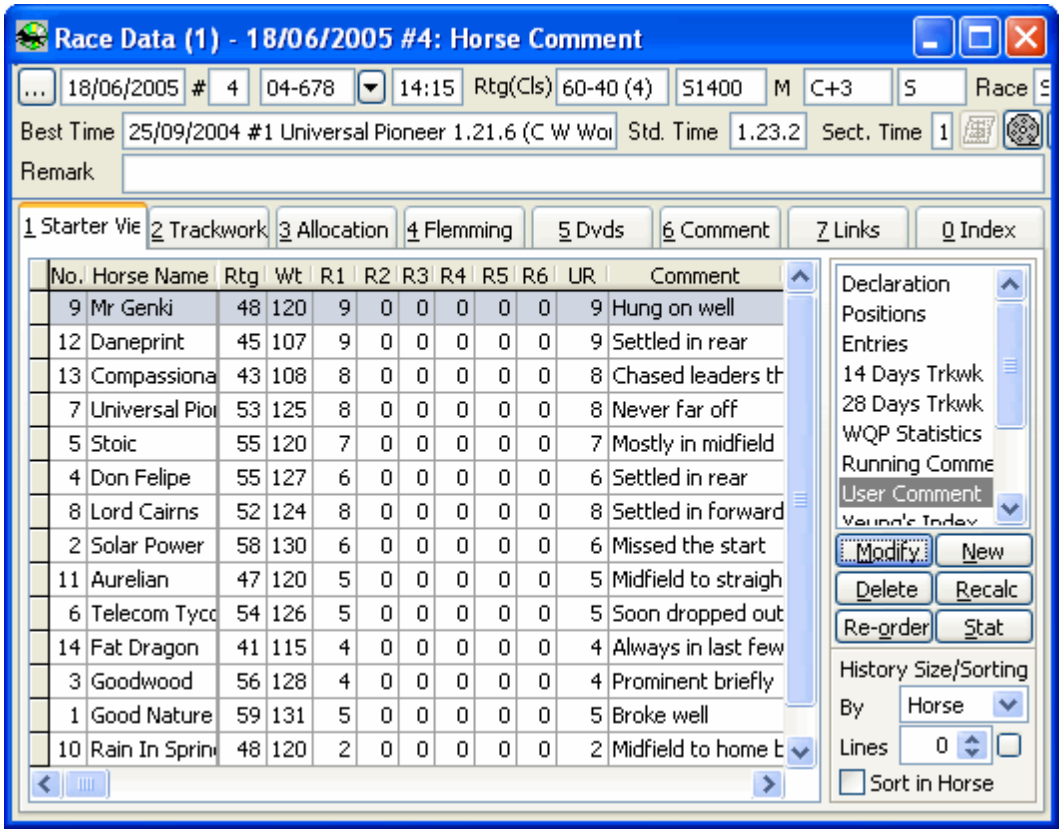
1. Open or create the view containing the data to be exported

In this example, we will open the built-in view "User Comments". If you have not made any change to the view, you will see a layout similar to the following figure. It will not be a problem if you have already made change to the view. You can proceed to the following step as long as the following columns still exist:

- No.
- Horse Name
- Rtg
- R1
- Comment

If any of the columns no longer exists, you may recreate it before proceeding into the following steps.

© Welkin Systems Limited


Race Data (1) - 18/06/2005 #4: Horse Comment
18/06/2005 # 4 04-678 14:15 Rtg(Cls) 60-40 (4) $1400 M C+3 S Race
Best Time 25/09/2004 #1 Universal Pioneer 1.21.6 (C W Woi Std. Time 1.23.2 Sect. Time 1
Remark
1 Starter Vie 2 Trackwork 3 Allocation 4 Flemming 5 Dvds 6 Comment 7 Links 0 Index
No. Horse Name Rtg Wt R1 R2 R3 R4 R5 R6 UR Comment
9 Mr Genki 48 120 9 0 0 0 0 0 9 Hung on well
12 Daneprint 45 107 9 0 0 0 0 0 9 Settled in rear
13 Compassiona 43 108 8 0 0 0 0 0 8 Chased leaders th
7 Universal Pior 53 125 8 0 0 0 0 0 8 Never far off
5 Stoic 55 120 7 0 0 0 0 0 7 Mostly in midfield
4 Don Felipe 55 127 6 0 0 0 0 0 6 Settled in rear
8 Lord Cairns 52 124 8 0 0 0 0 0 8 Settled in forward
2 Solar Power 58 130 6 0 0 0 0 0 6 Missed the start
11 Aurelian 47 120 5 0 0 0 0 0 5 Midfield to straigh
6 Telecom Tycc 54 126 5 0 0 0 0 0 5 Soon dropped out
14 Fat Dragon 41 115 4 0 0 0 0 0 4 Always in last few
3 Goodwood 56 128 4 0 0 0 0 0 4 Prominent briefly
1 Good Nature 59 131 5 0 0 0 0 0 5 Broke well
10 Rain In Sprin 48 120 2 0 0 0 0 0 2 Midfield to home t
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Modify New
Delete Recalc
Re-order Stat
History Size/Sorting
By Horse
Lines 0
Sort in Horse

2. Create columns of interest

The columns Season-#, FP, Top3 Rtg do not exist in the original built-in view. Create these 3 columns. Season-# and FP are built-in columns, you may simple choose them in the Built-In Item list. Please put the columns before the No. and Wt columns respectively. Create the user-defined column before the Wt column. Don't forget to set the column type to "Character" and "Chars/Digits" to 12 (see the figure below).

3. Define Stat Groups

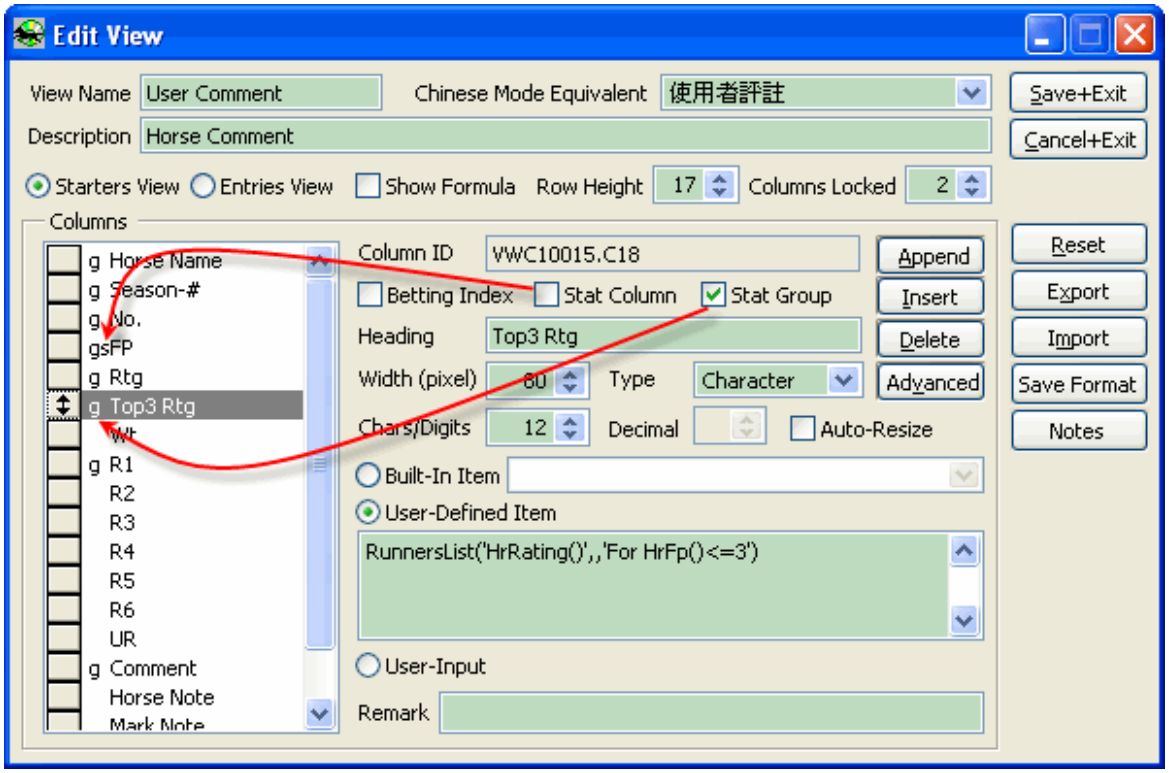
Set the following columns as "Stat Group". A letter "g" will be shown at the left of the column name in the column list:

- Horse Name
- Season-#
- No.
- FP
- Rtg
- R1
- Comment

4. Define Stat Column

Set the column FP as "Stat Column". A letter "s" will be show at the left of the column in the column list.

© Welkin Systems Limited




Note: Advance Data Export is actually using a statistical view to retrieve data from RaceMate's database. Even if we are not interested in any statistical item, we still need to set one. If we choose FP (finishing position) as the statistical column, the average, maximum and minimum values of FP of each data group (each horse in this example) will be shown in the data retrieved. If you are not interested in these figures, you may simply delete such columns in the resulting Excel sheet.

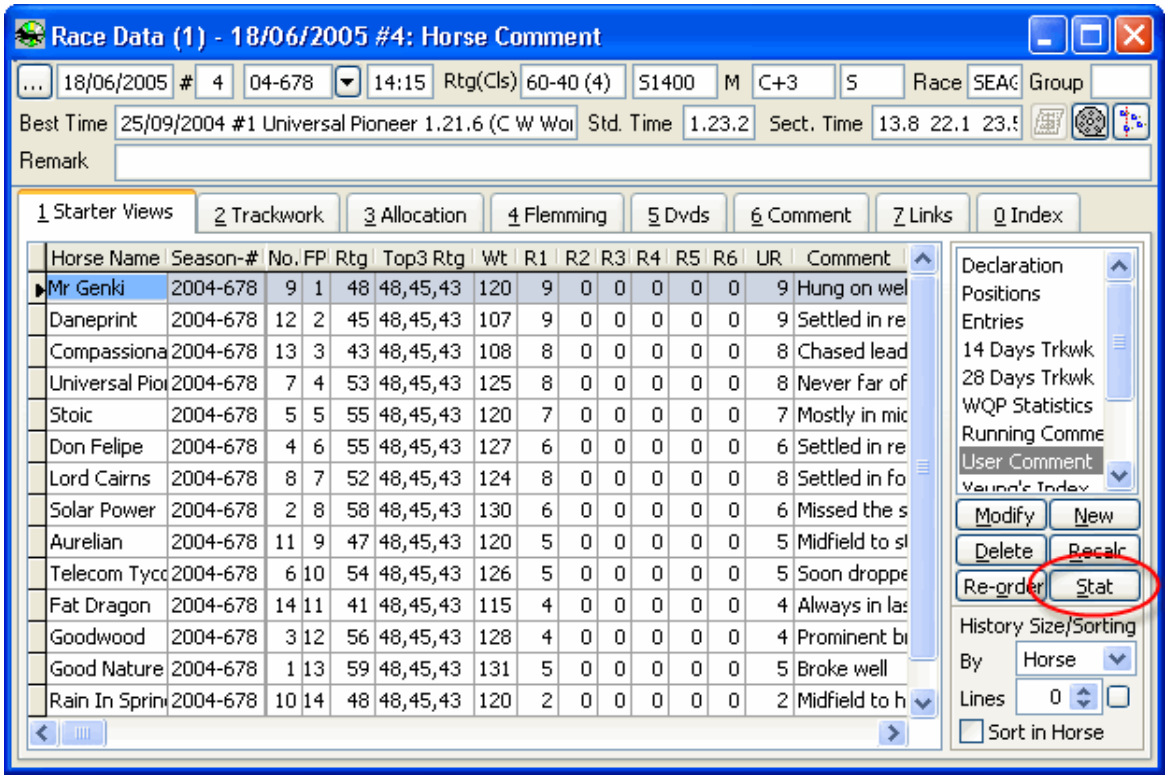
5. Rearrange Grouping Order

Move the Horse Name column to the top making it the first column. The purpose of doing this step is to make the final result be sorted firstly by Horse Name.

6. Save the View

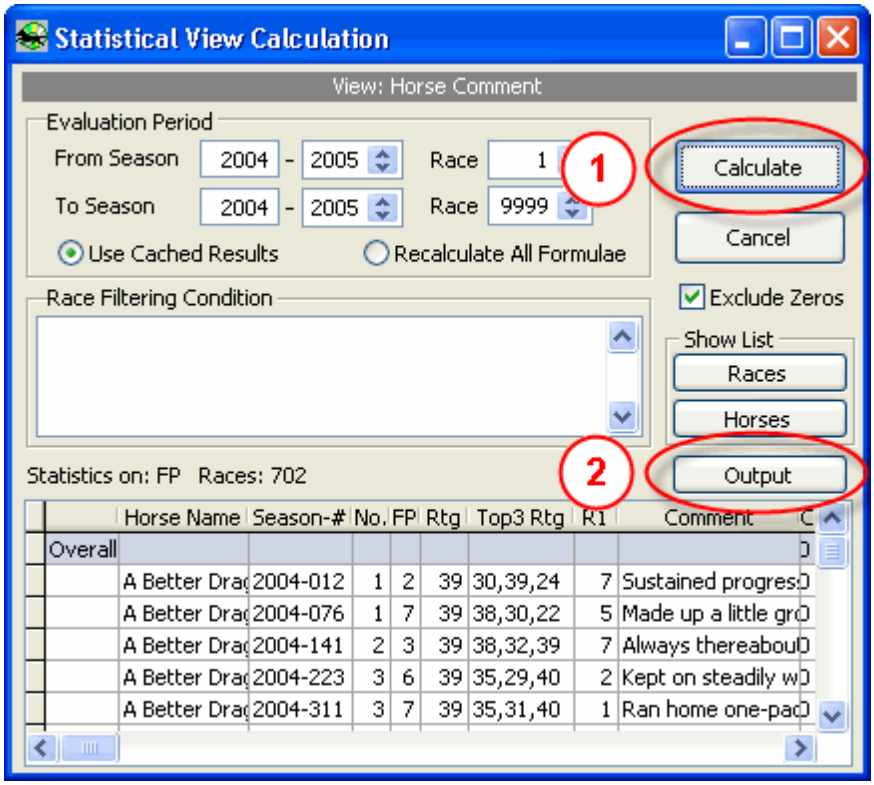
Click on the [Save+Exit] button to save the changes. You will see a screen similar to the following figure.

© Welkin Systems Limited


Race Data (1) - 18/06/2005 #4: Horse Comment
18/06/2005 # 4 04-678 14:15 Rtg(Cls) 60-40 (4) S1400 M C+3 S Race SEAG Group
Best Time 25/09/2004 #1 Universal Pioneer 1.21.6 (C W Wo Std. Time 1.23.2 Sect. Time 13.8 22.1 23.5
Remark
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Comment
7 Links
0 Index
Horse Name Season-# No. FP Rtg Top3 Rtg Wt R1 R2 R3 R4 R5 R6 UR Comment
Mr Genki 2004-678 9 1 48 48,45,43 120 9 0 0 0 0 0 9 Hung on wel
Daneprint 2004-678 12 2 45 48,45,43 107 9 0 0 0 0 0 9 Settled in re
Compassiona 2004-678 13 3 43 48,45,43 108 8 0 0 0 0 0 8 Chased lead
Universal Pio 2004-678 7 4 53 48,45,43 125 8 0 0 0 0 0 8 Never far of
Stoic 2004-678 5 5 55 48,45,43 120 7 0 0 0 0 0 7 Mostly in mic
Don Felipe 2004-678 4 6 55 48,45,43 127 6 0 0 0 0 0 6 Settled in re
Lord Cairns 2004-678 8 7 52 48,45,43 124 8 0 0 0 0 0 8 Settled in fo
Solar Power 2004-678 2 8 58 48,45,43 130 6 0 0 0 0 0 6 Missed the s
Aurelian 2004-678 11 9 47 48,45,43 120 5 0 0 0 0 0 5 Midfield to sl
Telecom Tyc 2004-678 6 10 54 48,45,43 126 5 0 0 0 0 0 5 Soon droppe
Fat Dragon 2004-678 14 11 41 48,45,43 115 4 0 0 0 0 0 4 Always in las
Goodwood 2004-678 3 12 56 48,45,43 128 4 0 0 0 0 0 4 Prominent b
Good Nature 2004-678 1 13 59 48,45,43 131 5 0 0 0 0 0 5 Broke well
Rain In Sprin 2004-678 10 14 48 48,45,43 120 2 0 0 0 0 0 2 Midfield to h
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Modify
New
Delete
Recalc
Re-order
Stat
History Size/Sorting
By Horse
Lines 0
Sort in Horse

7. Retrieve Data

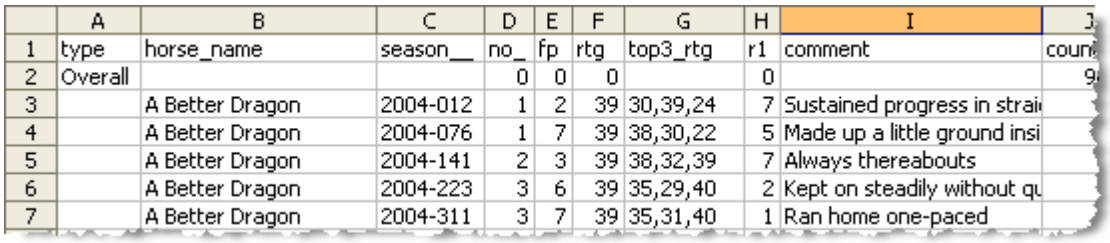
Click on the [Stat] button on the form in the above figure. You will see the following form. Set the evaluation period to races 1-9999 of the season 2004-2005. Click on [Calculate]. The data retrieved will be shown in the grid at the bottom.


Statistical View Calculation
View: Horse Comment
Evaluation Period
From Season 2004 - 2005 Race 1
To Season 2004 - 2005 Race 9999
Use Cached Results
Recalculate All Formulae
1
Calculate
Cancel
Exclude Zeros
Race Filtering Condition
Show List
Races
Horses
Output
2
Statistics on: FP Races: 702
Horse Name Season-# No. FP Rtg Top3 Rtg R1 Comment
Overall
A Better Dra 2004-012 1 2 39 30,39,24 7 Sustained progres
A Better Dra 2004-076 1 7 39 38,30,22 5 Made up a little gr
A Better Dra 2004-141 2 3 39 38,32,39 7 Always thereabou
A Better Dra 2004-223 3 6 39 35,29,40 2 Kept on steadily w
A Better Dra 2004-311 3 7 39 35,31,40 1 Ran home one-pac

© Welkin Systems Limited

8. Export result to Excel

Click on the [Output] button to send the retrieved data into an external file. If you choose Excel as the output format, a Microsoft Excel spreadsheet similar to the following will be created.



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | type | horse_name | season__ | no_ | fp | rtg | top3_rtg | r1 | comment | coun |
| 2 | Overall | | | 0 | 0 | 0 | | 0 | | 9 |
| 3 | | A Better Dragon | 2004-012 | 1 | 2 | 39 | 30,39,24 | 7 | Sustained progress in strai | |
| 4 | | A Better Dragon | 2004-076 | 1 | 7 | 39 | 38,30,22 | 5 | Made up a little ground insi | |
| 5 | | A Better Dragon | 2004-141 | 2 | 3 | 39 | 38,32,39 | 7 | Always thereabouts | |
| 6 | | A Better Dragon | 2004-223 | 3 | 6 | 39 | 35,29,40 | 2 | Kept on steadily without qu | |
| 7 | | A Better Dragon | 2004-311 | 3 | 7 | 39 | 35,31,40 | 1 | Ran home one-paced | |

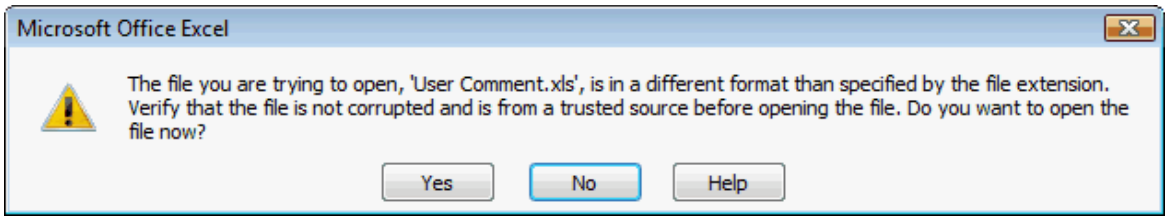
Note: Special characters in column headings are replaced by an underscore by RaceMate

9. Fine Tune Excel File

Finally you may delete the rows and columns that you don't need from the exported Excel sheet.

## 13.6 Format of Exported Excel Files

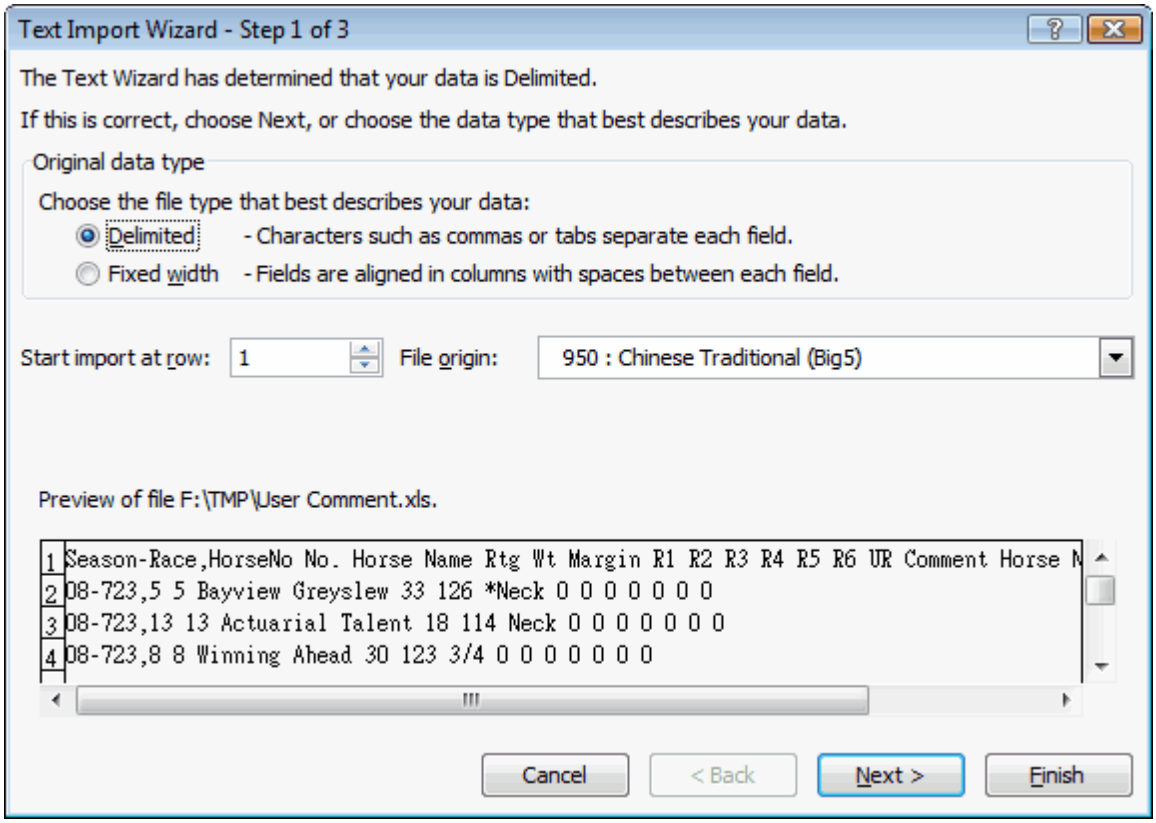
Most of the Excel files exported from RaceMate are actually tab-delimited plain text files with a file extension "xls". You may open the files in Excel by directly double-clicking on the files. If you are using Microsoft Excel 2007, you will see a message similar to the following. Such message will not appear in earlier versions of Excel.


Microsoft Office Excel
The file you are trying to open, 'User Comment.xls', is in a different format than specified by the file extension. Verify that the file is not corrupted and is from a trusted source before opening the file. Do you want to open the file now?
Yes
No
Help

Once you click on [Yes] the file will be opened.

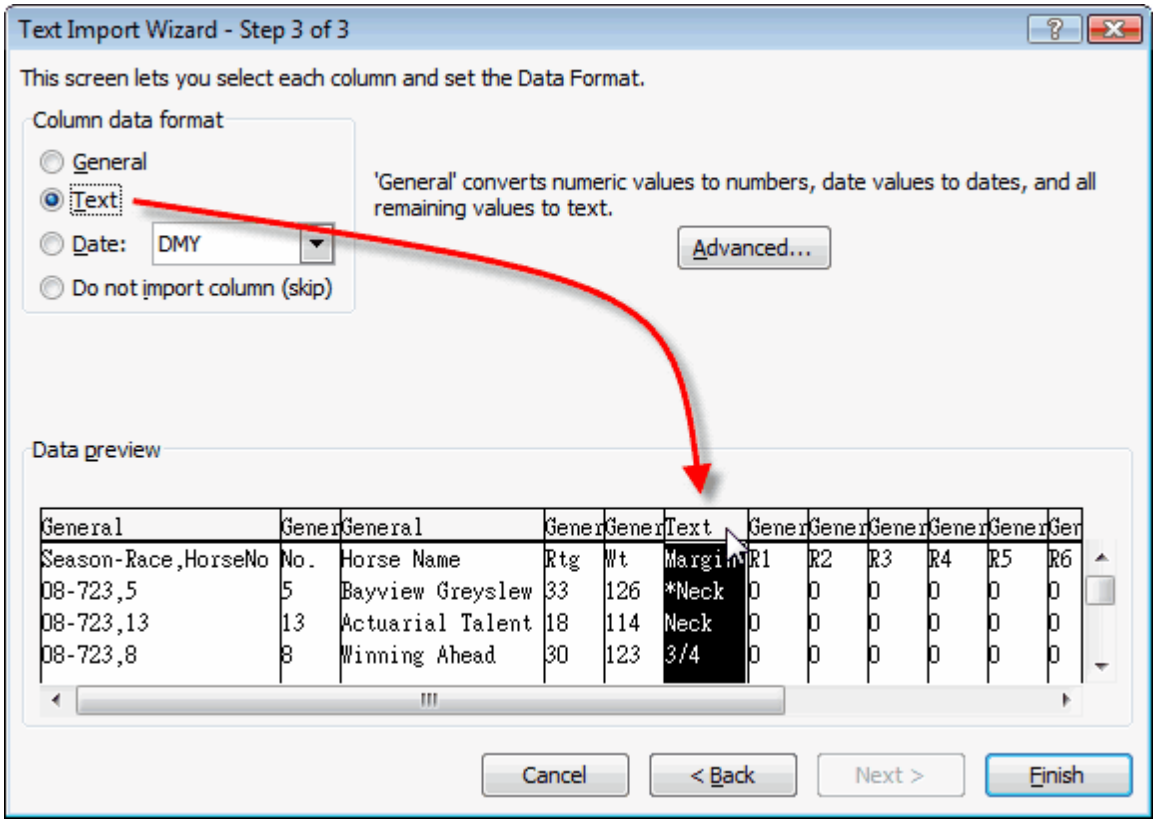
If you open the file via the Excel menu, you will see a form similar to the following.

© Welkin Systems Limited


Text Import Wizard - Step 1 of 3
The Text Wizard has determined that your data is Delimited.
If this is correct, choose Next, or choose the data type that best describes your data.
Original data type
Choose the file type that best describes your data:
Delimited - Characters such as commas or tabs separate each field.
Fixed width - Fields are aligned in columns with spaces between each field.
Start import at row: 1
File origin: 950 : Chinese Traditional (Big5)
Preview of file F:\TMP\User Comment.xls.
1 Season-Race,HorseNo No. Horse Name Rtg Wt Margin R1 R2 R3 R4 R5 R6 UR Comment Horse N
2 08-723,5 5 Bayview Greyslew 33 126 *Neck 0 0 0 0 0 0 0
3 08-723,13 13 Actuarial Talent 18 114 Neck 0 0 0 0 0 0 0
4 08-723,8 8 Winning Ahead 30 123 3/4 0 0 0 0 0 0 0
Cancel
< Back
Next >
Finish

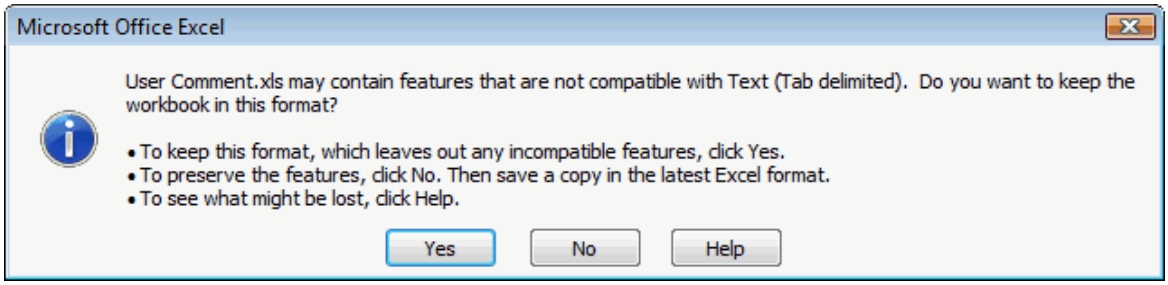
You may simply click on [Finish] to open the file. Generally speaking Excel will convert data into correct data types automatically. However sometimes Excel may do wrong conversions. For example, Excel always convert "margin" characters to dates. If you foresee any wrong data type conversions, you may click [Next] two times to proceed to the following data format setting form. The following example form shows how to tell Excel that this is a text data rather than dates.

© Welkin Systems Limited


Text Import Wizard - Step 3 of 3
This screen lets you select each column and set the Data Format.
Column data format
General
Text
Date: DMY
Do not import column (skip)
'General' converts numeric values to numbers, date values to dates, and all remaining values to text.
Advanced...
Data preview
General Gener General Gener Gener Text Gener Gener Gener Gener Gener Ger
Season-Race,HorseNo No. Horse Name Rtg Wt Margin R1 R2 R3 R4 R5 R6
08-723,5 5 Bayview Greyslew 33 126 *Neck 0 0 0 0 0 0
08-723,13 13 Actuarial Talent 18 114 Neck 0 0 0 0 0 0
08-723,8 8 Winning Ahead 30 123 3/4 0 0 0 0 0 0
Cancel
< Back
Next >
Finish

### Saving Exported Files

When you try to save a RaceMate exported file, you may see a message similar to the following. You may keep the files original text format or convert it to a true Excel file. Once you have converted it to a true Excel file, this file will not be able to be imported into a RaceMate on a computer without Microsoft Excel installed. Please also note that Excel 2007 files are in a special format which can't be read by earlier versions of Excel.

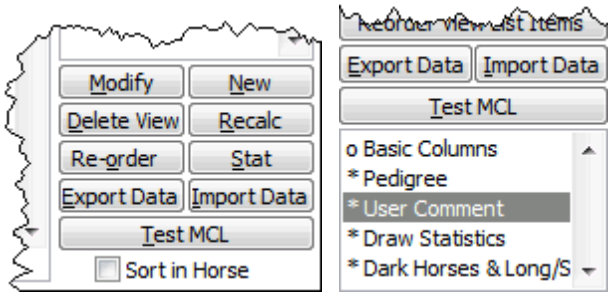

Microsoft Office Excel
User Comment.xls may contain features that are not compatible with Text (Tab delimited). Do you want to keep the workbook in this format?
• To keep this format, which leaves out any incompatible features, click Yes.
• To preserve the features, click No. Then save a copy in the latest Excel format.
• To see what might be lost, click Help.
Yes
No
Help

## 13.7 Importing View Data From Excel Files

### Importing View Data From Excel Files (or Tab-Delimited Files)

Click on the [Import Data] button to import data from from an Excel file into a view. If this file is a true Excel file, your computer must have Microsoft Excel installed. If this is a RaceMate exported file, Microsoft Excel is not required.

© Welkin Systems Limited




**Which Column will be Imported?**

RaceMate will import only the following columns that meet the following criteria.

1. The name of the Excel column exactly match the name of the view column
2. The column is a user input column

**Importing Files with a Primary Key**

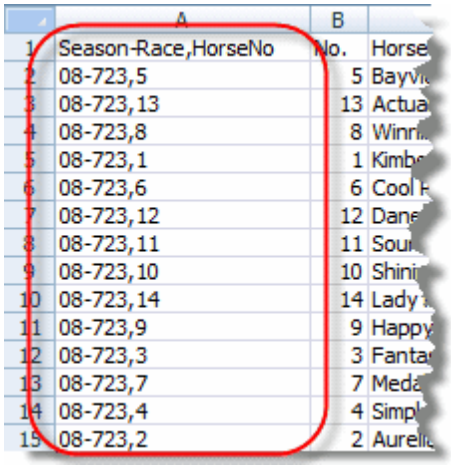
Although primary key is not necessary for importing, it has the following advantage.

- The sheet to be imported can contain data of many races.
- You can import data into races no matter they are currently displaying or not.

The heading of the primary key must be "Season-Race,HorseNo". The content of it should be as follows

<2-digit season><hyphen><season race index><horse no.>

Here is an example:



| | A | B | |
|---|---|---|---|
| 1 | Season-Race,HorseNo | No. | Horse |
| 2 | 08-723,5 | 5 | Bayv |
| 3 | 08-723,13 | 13 | Actua |
| 4 | 08-723,8 | 8 | Winn |
| 5 | 08-723,1 | 1 | Kimb |
| 6 | 08-723,6 | 6 | Cool H |
| 7 | 08-723,12 | 12 | Dane |
| 8 | 08-723,11 | 11 | Sou |
| 9 | 08-723,10 | 10 | Shini |
| 10 | 08-723,14 | 14 | Lady |
| 11 | 08-723,9 | 9 | Happy |
| 12 | 08-723,3 | 3 | Fanta |
| 13 | 08-723,7 | 7 | Meda |
| 14 | 08-723,4 | 4 | Simp |
| 15 | 08-723,2 | 2 | Aurel |

**Importing Files without a Primary Key**

- You can only import the data in the file into the view displaying on the active form.
- The import result will be affected by the row order of the current view.

**Data Merging**

© Welkin Systems Limited

RaceMate will import data in the Excel file and will not erase other data. That means you can tailor the lines in your Excel file to achieve data merging. That means you can update only a copy of continuous or non-continuous line of your view and leave other lines unchanged.

© Welkin Systems Limited

# Part XIV

# 14 System Maintenance and Configuration

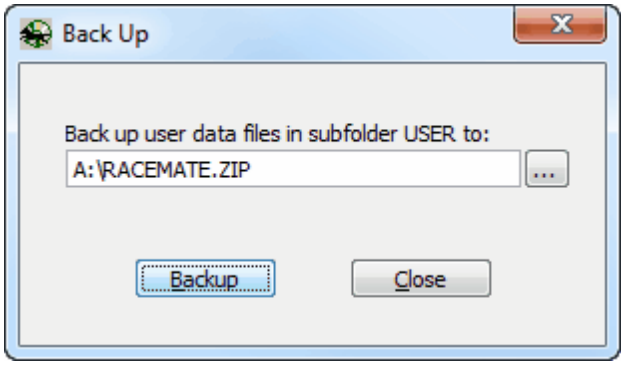
This chapter discusses how to change the default behaviours of RaceMate and how to maintain your user input in the database of RaceMate.

**>>> Subtopics of this section <<<**

- Backing Up and Restoring User Data
- RaceMate Notice
- System Configurations and Information
- Clearing Cache Folder
- Upgrading RaceMate or Installing Add-Ins

## 14.1 Backing Up and Restoring User Data

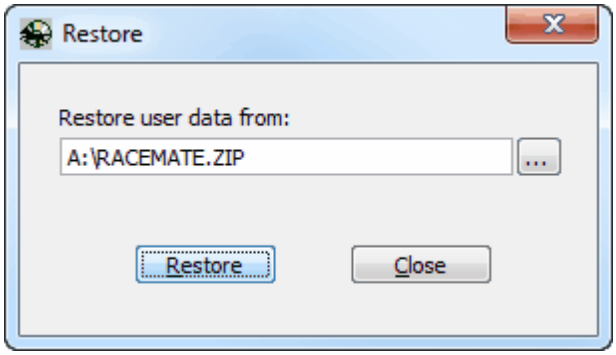
All user data (including user notes, ratings, conversion tables, and view definitions) of RaceMate are stored in the USER subfolder of RaceMate. You may back up your data by choosing [Back Up User Data Input and Changes Made by User] from the [Tools] menu of RaceMate. Put a floppy into drive A: (you may enter the path of a folder or a USB drive) and click on [Backup] to start back up. RaceMate will compress all files in the USER folder into a single file. The default file name is A:\RACEMATE.ZIP. If the floppy diskette is full, RaceMate will ask you to insert another one.




On the form above you may change the file name to any valid file name, e.g., E:\ABC\RACEMATE.RMB.

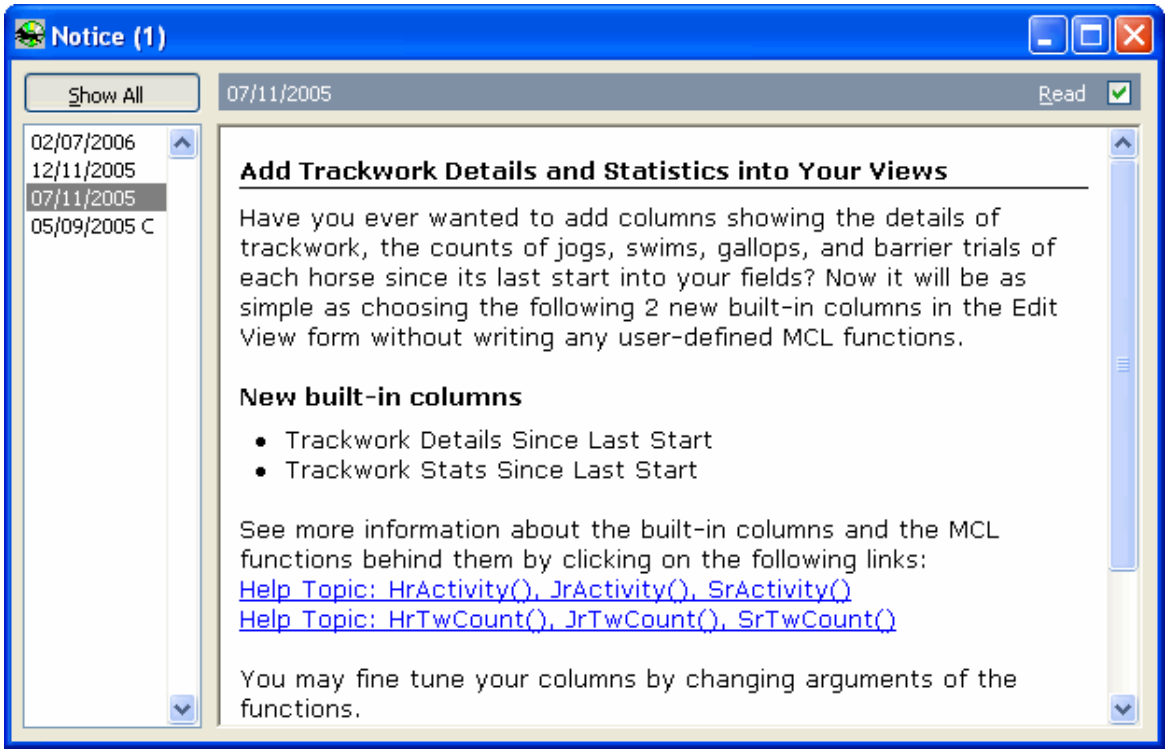
To restore data from your backup disk, choose [Restore User Data Input and Changes from Previous Backup] from the [Tools] menu of RaceMate. Insert your backup disk or select the path you have backed up your data to. Edit the file name to match the file name of your back up file, e.g., E:\ABC\RACEMATE.RMB. Click on [Restore] to start restoring data.

© Welkin Systems Limited




## 14.2 RaceMate Notice

Occasionally, you may see notices (figure below) which pops up on the screen when you start RaceMate. After you have read the notice, check the [Read] box at the right top of the Notice form to close the form. If you close the form without checking the [Read] box, next time when you start RaceMate, the notice will pop up once again on the screen.


Notice (1)
Show All
07/11/2005
Read
02/07/2006
12/11/2005
07/11/2005
05/09/2005 C
Add Trackwork Details and Statistics into Your Views
Have you ever wanted to add columns showing the details of trackwork, the counts of jogs, swims, gallops, and barrier trials of each horse since its last start into your fields? Now it will be as simple as choosing the following 2 new built-in columns in the Edit View form without writing any user-defined MCL functions.
New built-in columns
Trackwork Details Since Last Start
Trackwork Stats Since Last Start
See more information about the built-in columns and the MCL functions behind them by clicking on the following links:
Help Topic: HrActivity(), JrActivity(), SrActivity()
Help Topic: HrTwCount(), JrTwCount(), SrTwCount()
You may fine tune your columns by changing arguments of the functions.

If you have declared 'Read' on a notice, you can still get access to the notice again by choosing the [Notice] option from the [Help] menu open the Notice form. You will see a list of notice dates on the left hand side of the form. Click on any of them to show the notice content. Please note that if there are any notices which are still marked as unread, only the date of these unread notices will be shown on the list. However, you may click on [Show All] to show all notices.

© Welkin Systems Limited

## 14.3 System Configurations and Information

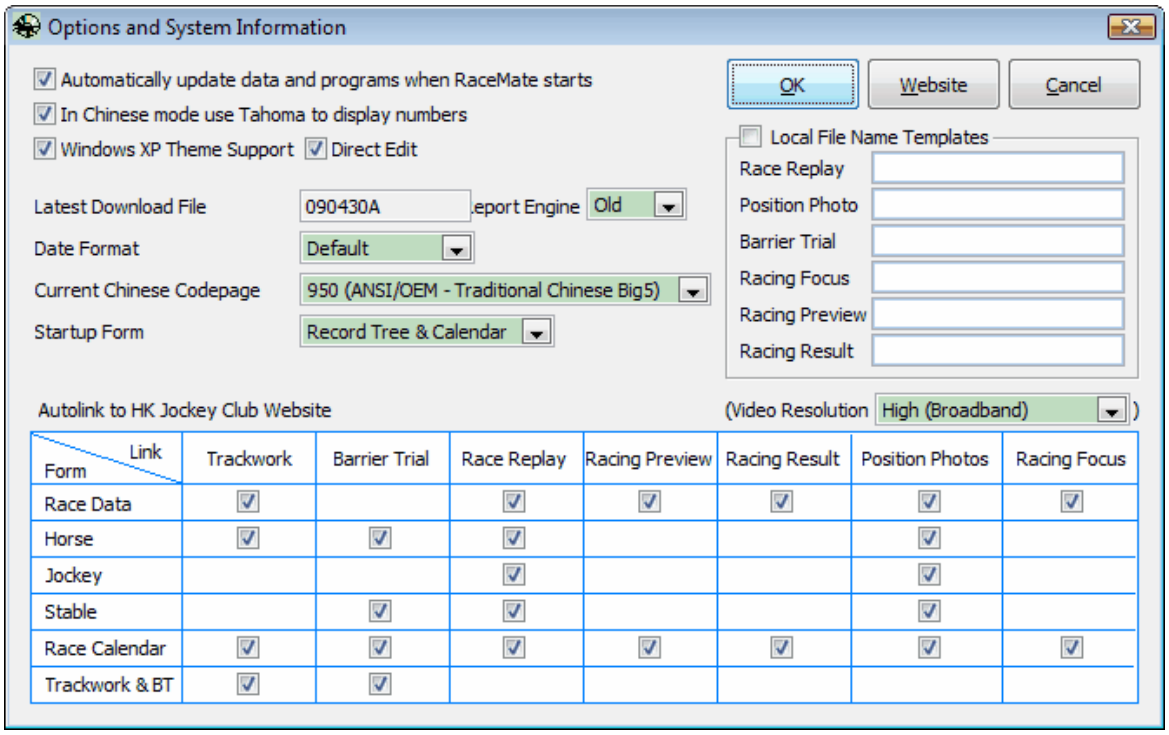
This section will discuss how to get information about the system and how to change the system settings.

**>>> Subtopics of this section <<<**

- Options and System Information
- About RaceMate

### 14.3.1 Options and System Information

Choose [Options and System Information] from the [Tools] menu to open the following form for viewing and define system settings:




**Phone No. of RaceMate Data Server**

If you have selected Dial-Up Connection as your download method, RaceMate will dial the number in this box to connect to RaceMate's data server. If the location where your RaceMate is installed requires special phone number prefix or you need to add an international call number, here is the place to put the extra numbers.

**Latest Download File**

This field shows the name of the latest data update file that you downloaded and processed.

**Download Settings**

© Welkin Systems Limited

Open the DownloadSettings form.

**Date Format**

You may choose a date format here. Available options are listed in the following table.

| Format | Description | Example |
|---|---|---|
| DD/MM/YYYY | Day/Month/Year | 26/03/2004 |
| MM/DD/YYYY | Month/Day/Year | 03/26/2004 |
| YYYY/MM/DD | Year/Month/Day | 2004/03/26 |
| Default | Follow the format settings Windows Control Panel | |

**Current Chinese Codepage**

RaceMate will automatically detect the codepage of your Windows and properly display Chinese characters in either Traditional or Simplified characters. However, if you are using an external Chinese characters display system such as RichWin and China Star, you may need to manually set your current codepage here.

**Windows XP Theme Support**

If you are using Windows XP, you may turn off or turn on Windows XP Theme Support here.

**Direct Edit**

If you select this option, you can input data into the editing controls without first clicking on the [Edit] button of the toolbar. See Editing Data for details.

**Autolinks to HK Jockey Club Website**

RaceMate will put hyperlinks to the multimedia resources of the HK Jockey Club Website into records of Race, Horse, Jockey, Stable and Track Work. If you are not interested in some of the links and would like to remove them from the record forms, you may deselect them here.

**Connection Speed**

If you are using a broadband Internet connection, you should choose [Broadband Connection] here to connect to the high quality multimedia video files of the HK Jockey Club. Otherwise, you should choose [56K Dial-Up Connection] to connect to a trimmed-down version of the files.

### 14.3.2 About RaceMate

Select [About RaceMate] from the [Help] menu. A form showing the program information will be shown as follows. You may click on

1. the [Program Files Details] button to show a list of program files
2. the program filter link to open the program folder

© Welkin Systems Limited

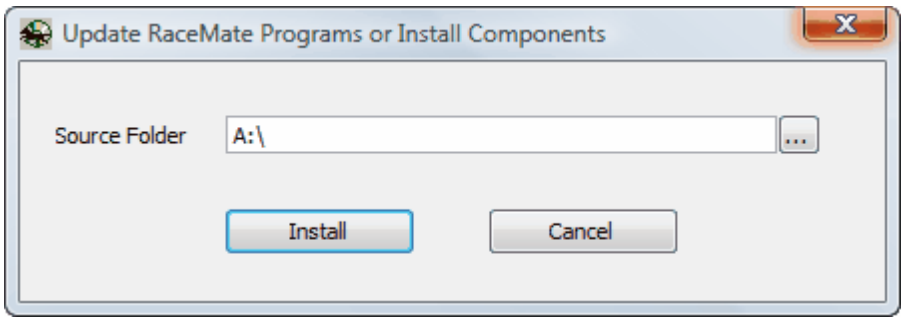
## 14.4 Clearing Cache Folder

Whenever you find that the data in the cache folder is out-of-data, please select [Clean Up Calculation Results in Cache] to clear the data in the cache folder.

Please see Cache for View Data for functions about the cache folder.

## 14.5 Upgrading RaceMate or Installing Add-Ins

Whenever you have got either a RaceMate update file or an version upgrade file, please select [Upgrading RaceMate or Installing Add-Ins] from the [Tools] menu, you will see the following dialogue box. If your file is not in drive A:, click on the small button at the right of the source folder text box to select the correct folder. Then click on [Install] to install the file.




© Welkin Systems Limited

# Part XV

# 15 MCL Tutorial

Model Construction Language (MCL)，Model Construction Language (MCL) is a programming language that comes with RaceMate for retrieving information from the database of RaceMate. Please download an MCL Tutorial from the following web site:

http://www.racematenet.com/english/download_e.htm

You will be able to put MCL into use in a short period of time after going through the tutorial.

© Welkin Systems Limited

# Part XVI

# 16 MCL Reference

Model Construction Language (MCL) is a programming language that comes with RaceMate for retrieving information from the database of RaceMate.

Most MCL functions come in three versions:

1. The Horse Record version (function names begin with Hr), which retrieves data or perform statistics in past records of horses
2. The Jockey Record version (function names begin with Jr), which retrieves data or perform statistics in past records of jockeys
3. The Stable Record version (function names begin with Sr), which retrieves data or perform statistics in past records of stables

For example, the following functions return the finishing position of a horse, a jockey, and a stable in last 1650m races respectively: (Please note that the last 1650m race of a jockey and the last 1650m race of a horse may be 2 different races.)

```
HrFP(1,,'HrDistance( )=1650')
JrFP(1,,'HrDistance( )=1650')
SrFP(1,,'HrDistance( )=1650')
```

If we are going to retrieve data of the underlying race, i.e., the race displaying on the screen currently, basically, there is no difference among the three versions. That is, the following functions are exchangeable:
HrFP( ), JrFP( ), SrFP( )  or
HrFP(0), JrFP(0), SrFP(0)

Similarly the following 2 functions will return identical data: (The functions retrieve the finishing position of a horse in its last 1650m start.)

```
HrFP(1,,'HrDistance( )=1650')
HrFP(1,,'JrDistance( )=1650')
```

**>>> Subtopics of this section <<<**


- Arguments and Variables of MCL functions
- Null Values
- Information about a Start
- Static information of Horse, Jockey, and Stable
- Track Work Information
- Race and Meeting Information
- Data Collection Assistant Functions
- Statistics on Past Records
- Statistics in a Race
- Arithmetic Functions
- String Functions
- List Functions
- Conversion Functions
- Multimedia & Internet Functions
- Miscellaneous Functions
- VFP Functions & Operators


© Welkin Systems Limited

# 16.1 Arguments and Variables of MCL functions

An argument is a value that determines the returned value of a function. This chapter will show you how arguments are used in MCL functions.

**>>> Subtopics of this section <<<**

- Optional Arguments
- Data Types of Arguments
- The nBackRaces Argument of an MCL function
- The fCondition Argument (MCL Condition)
- When to use the While keyword
- Quoted MCL Expression Arguments
- The DefaultVal Argument
- The cIncluding Argument

## 16.1.1 Optional Arguments

Most arguments of MCL functions are optional. Required arguments are underlined in all syntax descriptions in this reference. For example, in the following function, all arguments are optional except nMCLExpr.

```
HrSum('nMCLExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

If the arguments that you are going to omit are the rightmost ones, the commas between the arguments can be taken away. For example, the following two expressions are basically identical:

```
HrSum('HrWinTickets( )',3)
HrSum('HrWinTickets( )',3,,,)
```

If the arguments that you are going to omit are not the rightmost ones, you must keep the commas, for example:

```
HrSum('HrWinTickets( )',,'For HrFP( )=1')
```

For Visual FoxPro built-in functions, you can only omit rightmost optional parameters but not the ones in the middle. For example, the following function will result in error:

```
Trim('ABC. ',,[ ],[.])
```

You must rewrite it as:

```
Trim('ABC. ',0,[ ],[.])
```

## 16.1.2 Data Types of Arguments

The data types required by the arguments of an MCL functions are specified by the following prefixes:

| Prefix | Data type required | Example |
|---|---|---|
| n | Numeric | nRaces |
| c | Character | cIncluding |
| f | MCL Condition | 'fCondition' |
| d | Date | dDate |

© Welkin Systems Limited

| l | Logical | lTrueOrFalse |
|---|---|---|
| e | Arguments that accept more than one data type | eDefaultVal |

### 16.1.3 The nBackRaces Argument of an MCL function

Most MCL functions retrieving horse or running information accept an nBackRaces argument. This argument tells RaceMate datum of which race in the past record is to be retrieved. For example, the HrBodyweight( ) function retrieves the bodyweight of a horse as follows:

| Function & Argument | Datum Retrieved |
|---|---|
| HrBodyweight( ) or HrBodyweight(0) | Body weight of the horse in the underlying race will be retrieved. (Underlying race means the race that the function applying to.) |
| HrBodyweight(1) | Body weight of the horse in last running of the horse will be retrieved |
| HrBodyweight(2) | Body weight of the horse in running prior to last running will be retrieved |

### 16.1.4 The fCondition Argument (MCL Condition)

Most MCL functions accept an MCL condition argument. Here are 2 examples:

```
HrSum('HrWinTickets( )',,'For HrFP( )=1')
```

In the example above, the MCL condition 'For HrFP( )=1' tells RaceMate to sum the Win Investments of the horse in only in the winning starts, i.e., HrFP( )=1.

```
HrRating(1,'For HrFP( )=1')
```

In the example above, the MCL condition 'For HrFP( )=1' tells RaceMate to show the rating of the last (the first argument: 1) winning (the MCL condition: 'For HrFP( )=1') start .

An MCL condition must be quoted. You may use single quotes (' '), double quotes (" ") or square brackets ([ ]) provided that the quotes do not already exist in the condition expression.

An MCL Condition is composed of a For clause and a While clause. While RaceMate is scanning backwards to retrieve data or do statistics on past records:

- a For clause asks RaceMate to act on only the records that satisfies the logical condition that follows the For keyword.
- a While clause causes the MCL function to stop action once the logical expression that follows the While keyword evaluates to false.

Both For and While clauses are optional. See the topic When to use the While keyword for the difference between For and While.

The conditions specified by the For and While keyword can be compound conditions, i.

© Welkin Systems Limited

e. criteria connected by And and Or, for example:

```
HrSum('HrWinTickets( )',,'For HrFP( )=1 And HrDistance( )=1400')
```

**Important:** If both For and While clauses exist in the condition argument, please place the For clause before the While clause.

### 16.1.5 When to use the While keyword

The following HrAverage('HrRating( )') returns the average Jockey Club Rating of a horse.

```
HrAverage('HrRating( )')
```

When there is no MCL conditions, the function will do the average on the whole set of past records. If you want to get the average rating of the horse in only the races within the last 180 days (HrDaysAgo( )<=180), you may rewrite the function as follows:

```
HrAverage('HrRating( )',,'For HrDaysAgo( )<=180')
```

However, in the MCL condition expression of the example above, only For is used. When there is only For and no While in an MCL condition, RaceMate will scan all history records to filter out those satisfying the For criterion. You will find that rewriting the above function as follows will yield a much faster calculation speed.

```
HrAverage('HrRating( )',,'While HrDaysAgo( )<=180')
```

RaceMate will scan the past records backwards one by one. Once RaceMate finds the record does not meet the criterion specified by the While condition, scanning will stop and result will be returned immediately.

**Important:** If both For and While clauses exist in the condition argument, please place the For clause before the While clause.

### 16.1.6 Quoted MCL Expression Arguments

An MCL expression is an expression composed of one or more MCL functions. Most statistical MCL functions accept Quoted MCL Expression arguments. Usually single quotes are used. However, the single quotes may be replaced with double quotes (" ") or square brackets ([ ]) provided that the quotes do not already exists inside the MCL expression.

The first argument of the following HrSum( ) function is an example:

```
HrSum('HrWinTickets( )',3)
```

The above function retrieves the sum of Win Investments on the horse in the past 3 runnings. The MCL expression HrWinTicks( ) is quoted and passed into the function. The function can be rewritten as:

```
HrSum("HrWinTickets( )",3)
```

or

```
HrSum([HrWinTickets( )],3)
```

Below is another example that summing up both Win Investments and Place

© Welkin Systems Limited

Investments of the horse in last 3 runnings. In the following example, the MCL expression contains two MCL functions:

```
HrSum('HrWinTickets( )+HrPlaceTickets( )',3)
```

### 16.1.7 The DefaultVal Argument

Many MCL functions accept a DefaultVal argument. If the target race cannot be found, the value passed to the DefaultVal argument will be returned. For example, the syntax of the HrBodyWeight( ) function is as follows:

```
HrBodyWeight(nBackRaces,'fCondition',nDefaultVal)
```

The following function call returns the body weight of the horse in its last start. If last start does not exist at all, 1050 pounds will be returned:

```
HrBodyWeight(1,,1050)
```

### 16.1.8 The cIncluding Argument

A jockey may take part in more than one race in a meeting. A stable may start more than one horse in a race. When we are doing analysis on a race, especially a coming race, usually we do not want data of other races of the same day to be included. For example, we are doing statistics for a jockey in the 6 th race of the meeting on 20/3/2008, usually we do not want data of other races of the same meeting, say the 4 th race of in the same day, to be counted. The cIncluding parameter can control whether races in the same meeting are taken into account.

| cIncluding | Description |
|---|---|
| | If you do not pass any value to the cIncluding parameter, races in the same meeting will not be taken into account by default. |
| 'D' | Starts of the same meeting will be taken into account. However, starts in the same race will still be excluded. 'D' is not applicable to horse records because a horse will not start twice in one meeting. |
| 'R' | Starts in the same meeting including those in the same race will be taken into account. 'R' is only applicable to stable records because only stable can have 2 starts in the same race. |
| 'C' (only for statistics) | Including the underlying start, i.e., all horses in the same day, same race and underlying start. Only valid for statistical functions. |

For example, the average rating figure returned by the following functions will be different.

| | |
|---|---|
| SrAverage('SrRating( )',20) | Starts in the same meeting are not counted |
| SrAverage('SrRating( )',20,,'D') | Starts in the same meeting, excluding the same race, will be counted. |
| SrAverage('SrRating( )',20,,'R') | Starts in the same meeting, including the same race, will be counted. |

© Welkin Systems Limited

# 16.2 Null Values

Null means anticipated data are unavailable. Null is extensively used in RaceMate views. The chapter will discuss how Null is used in RaceMate.

**>>> Subtopics of this section <<<**

- Characteristics of Null Values
- Absence of Values
- Null Values in Statistics
- Propagation of Null Values
- Errors in MCL Formulas

## 16.2.1 Characteristics of Null Values

RaceMate MCL provides support for null values. This support simplifies the task of representing unknown data. Null values are:

- Equal to the absence of any value.
- Different than zero, the empty string (""), or blank.
- Sorted ahead of other data.
- Propagated in calculations and most functions.

A null value is displayed as .NULL. When you double-clicking on a cell of null value in a RaceMate view, RaceMate will display a box showing information, if available, about how the value is got.

## 16.2.2 Absence of Values

The function HrRating(1) will normally return the rating of a horse of its previous start. However, for a new horse without any previous start, a null value will be returned.

## 16.2.3 Null Values in Statistics

In statistical MCL functions, null values will not be counted. For example, when we apply the Ave FP model (see MCL Tutorial) to the the horse Cyber Gulch in the 633 rd race of season 1999-2000, we will get the .NULL. result. It is because at the time of the race, the horse Cyber Gulch has only started once. In that start, its finishing position was 13. Thus, the function HrFP(1) returns 13 (column FP(1)) and function HrFP(2) returns .NULL.

When a statistical MCL function finds no data to process, the function will return a null value as well.

## 16.2.4 Propagation of Null Values

Whenever a formula contains an item of null value, the formula always gives a result of .NULL. except that the null value is the argument of the function IsNull( ) or the first argument of NVL( ).

For example, if you write a formula:

```
14 - Column('Ave FP')
```

© Welkin Systems Limited

and the value of the item Column('Ave FP') is .NULL., the result yielded from the formula will be .NULL.

i.e. 14 - .NULL. = .NULL.

### 16.2.5 Errors in MCL Formulas

Whenever errors are found in an MCL formula, the formula will yield a null value. You may double-click on the cell of the null value to see a description of the error.

## 16.3 Information about a Start

This group of functions returns basic information about a start.

**>>> Subtopics of this section <<<**

- HrAge( ), JrAge( ), SrAge( )
- HrAgeMonths( ), JrAgeMonths( ), SrAgeMonths( )
- HrAllowance( ), JrAllowance( ), SrAllowance( )
- HrBleedDate( ), JrBleedDate( ), SrBleedDate( )
- HrBleedDateStr( ), JrBleedDateStr( ), SrBleedDateStr( )
- HrBlinker( ), JrBlinker( ), SrBlinker( )
- HrBlinkerAbbr( ), JrBlinkerAbbr( ), SrBlinkerAbbr( )
- HrBlinkerAbbrNew( ), JrBlinkerAbbrNew( ), SrBlinkerAbbrNew( )
- HrBlinkerNew( ), JrBlinkerNew( ), SrBlinkerNew( )
- HrBodyWeight( ), JrBodyWeight( ), SrBodyWeight( )
- HrClass( ), JrClass( ), SrClass( )
- HrClassID( ), JrClassID( ), SrClassID( )
- HrCurRating( ), JrCurRating( ), SrCurRating( )
- HrDraw( ), JrDraw( ), SrDraw( )
- HrEntriesWeight( ), JrEntriesWeight( ), SrEntriesWeight( )
- HrFavorite( ), JrFavorite( ), SrFavorite( )
- HrFinalOdds( ), JrFinalOdds( ), SrFinalOdds( )
- HrFP( ), JrFP( ), SrFP( )
- HrFPX( ), JrFPX( ), SrFPX( )
- HrHorseNo( ), JrHorseNo( ), SrHorseNo( )
- HrHorseNoStr( ), JrHorseNoStr( ), SrHorseNoStr( )
- HrIsRunner( ), JrIsRunner( ), SrIsRunner( )
- HrLastSectPos( ), JrLastSectPos( ), SrLastSectPos( )
- HrLoadDiff( ), JrLoadDiff( ), SrLoadDiff( )
- HrLoadDiffStr( ), JrLoadDiffStr( ), SrLoadDiffStr( )
- HrMargin( ), JrMargin( ), SrMargin( )
- HrMarginLen( ), JrMarginLen( ), SrMarginLen( )
- HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( )
- HrNetLoadDiff( ), JrNetLoadDiff( ), SrNetLoadDiff( )
- HrOdds( ), JrOdds( ), SrOdds( )
- HrOdds1Day( ), JrOdds1Day( ), SrOdds1Day( )
- HrOdds5Min( ), JrOdds5Min( ), SrOdds5Min( )
- HrOverweight( ), JrOverweight( ), SrOverweight( )
- HrPastIncident( ), JrPastIncident( ), HrPastIncident( )
- HrPlaceTickets( ), JrPlaceTickets( ), SrPlaceTickets( )
- HrPositions( ), JrPositions( ), SrPositions( )

© Welkin Systems Limited

- HrPrivilege( ), JrPrivilege( ), SrPrivilege( )
- HrPrivilegeID( ), JrPrivilegeID( ), SrPrivilegeID( )
- HrRunning( ), JrRunning( ), SrRunning( )
- HrRacingWeight( ), JrRacingWeight( ), SrRacingWeight( )
- HrRating( ), JrRating( ), SrRating( )
- HrRatingDiff( ), JrRatingDiff( ), SrRatingDiff( )
- HrReserved( ), JrReserved( ), SrReserved( )
- HrSeconds( ), JrSeconds( ), SrSeconds( )
- HrSectionPos( ), JrSectionPos( ), SrSectionPos( )
- HrSectMargins( ), JrSectMargins( ), SrSectMargins( )
- HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( )
- HrSectMargLensX( ), JrSectMargLensX( ), SrSectMargLensX( )
- HrSectPositions( ), JrSectPositions( ), SrSectPositions( )
- HrSectTime( ), JrSectTime( ), SrSectTime( )
- HrSectTimew( ), JrSectTimew( ), SrSectTimew( )
- HrSpecialBit( ), JrSpecialBit( ), SrSpecialBit( )
- HrSpecialBitAbbr( ), JrSpecialBitAbbr( ), SrSpecialBitAbbr( )
- HrSpeed( ), JrSpeed( ), SrSpeed( )
- HrStablePrefence( ), JrStablePrefence( ), SrStablePrefence( )
- HrStartID( ), JrStartID( ), SrStartID( )
- HrStartIDOn(), JrStartIDOn(), SrStartIDOn()
- HrStakes( ), JrStakes( ), SrStakes( )
- HrTime( ), JrTime( ), SrTime( )
- HrTimew( ), JrTimew( ), SrTimew( )
- HrUpdown( ), JrUpdown( ), SrUpdown( )
- HrUpdownID( ), JrUpdownID( ), SrUpdownID( )
- HrWeight( ), JrWeight( ), SrWeight( )
- HrWinTickets( ), JrWinTickets( ), SrWinTickets( )
- HrWQPStat( ), JrWQPStat( ), SrWQPStat( )

### 16.3.1 HrAge( ), JrAge( ), SrAge( )

The functions return the age of the horse at the time it runs.

#### Syntax

```
HrAge(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrAge(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrAge(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric

### 16.3.2 HrAgeMonths( ), JrAgeMonths( ), SrAgeMonths( )

The functions return the age of the horse in number of months at the time it runs.

#### Syntax

```
HrAgeMonths(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrAgeMonths(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrAgeMonths(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

© Welkin Systems Limited

Numeric

### 16.3.3 HrAllowance( ), JrAllowance( ), SrAllowance( )

Return the weight allowance for a jockey, including apprentice allowance and freelance allowance.

#### Syntax

```
HrAllowance(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrAllowance(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrAllowance(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric

### 16.3.4 HrBleedDate( ), JrBleedDate( ), SrBleedDate( )

The functions return the bleeding date, if any, of the horse.

#### Syntax

```
HrBleedDate(nBackRaces,'fCondition',dDefaultVal)
JrBleedDate(nBackRaces,'fCondition',dDefaultVal)
SrBleedDate(nBackRaces,'fCondition',dDefaultVal)
```

#### Returns

Date

### 16.3.5 HrBleedDateStr( ), JrBleedDateStr( ), SrBleedDateStr( )

The functions is similar to the HrBleedDate( ), JrBleedDate( ) and SrBleedDate( ) functions except that this function return bleeding dates character strings.

#### Syntax

```
HrBleedDateStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBleedDateStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBleedDateStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=10)

### 16.3.6 HrBlinker( ), JrBlinker( ), SrBlinker( )

The functions return the blinker information of the horse.

© Welkin Systems Limited

**Syntax**

```
HrBlinker(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBlinker(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBlinker(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=44 in English language mode, Width=26 in Chinese language mode)

### 16.3.7 HrBlinkerAbbr( ), JrBlinkerAbbr( ), SrBlinkerAbbr( )

The functions return the blinker information of the horse using the coding system of the Hong Kong Jockey Club.

**Syntax**

```
HrBlinkerAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBlinkerAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBlinkerAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=3)

### 16.3.8 HrBlinkerAbbrNew( ), JrBlinkerAbbrNew( ), SrBlinkerAbbrNew( )

If this is the first time the horse wears this equipment, the functions return the Jockey Club abbreviation of the equipment. Otherwise an empty string will be returned.

**Syntax**

```
HrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=3)

### 16.3.9 HrBlinkerNew( ), JrBlinkerNew( ), SrBlinkerNew( )

If this is the first time the horse wears this equipment, the functions returns the name of the equipment. Otherwise an empty string will be returned.

**Syntax**

```
HrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBlinkerNew(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=44 in English language mode, Width=26 in Chinese language mode)

© Welkin Systems Limited

### 16.3.10 HrBodyWeight( ), JrBodyWeight( ), SrBodyWeight( )

The functions return the body weight of the horse.

#### Syntax

```
HrBodyWeight(nBackRaces,'fCondition',nDefaultVal)
JrBodyWeight(nBackRaces,'fCondition',nDefaultVal)
SrBodyWeight(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

### 16.3.11 HrClass( ), JrClass( ), SrClass( )

The functions return the class of the horse in the start. The following table shows the returned values:

| Class | Returned value in Chinese Mode | Returned value in English Mode |
|---|---|---|
| New | 新 | New |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 5 |
| 6 | 6 | 6 |

#### Syntax

```
HrClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=3)

#### See Also

HrRaceClass( ), HrClassID( )

### 16.3.12 HrClassID( ), JrClassID( ), SrClassID( )

The functions return the class ID of the horse in the start.

| Class | Returned value |
|---|---|

© Welkin Systems Limited

| | |
|---|---|
| New | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| Premium | 8 |
| Open | 9 |

**Syntax**

```
HrClassID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrClassID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrClassID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

**See Also**

HrClass( ), HrRaceClass( )

### 16.3.13 HrCurRating( ), JrCurRating( ), SrCurRating( )

The functions return the current (latest) Jockey Club rating of the horse.

**Syntax**

```
HrCurRating()
JrCurRating(nBackRaces,'fCondition',nDefaultVal)
SrCurRating(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**See Also**

HrRating( ), HrRaceRating( )

### 16.3.14 HrDraw( ), JrDraw( ), SrDraw( )

The functions return the draw number of the horse.

**Syntax**

```
HrDraw(nBackRaces,'fCondition',nDefaultVal)
JrDraw(nBackRaces,'fCondition',nDefaultVal)
SrDraw(nBackRaces,'fCondition',nDefaultVal)
```

© Welkin Systems Limited

**Returns**

Numeric

### 16.3.15 HrEntriesWeight( ), JrEntriesWeight( ), SrEntriesWeight( )

Loading of the horse printed on the Entries List

**Syntax**

```
HrEntriesWeight(nBackRaces,'fCondition',nDefaultVal)
JrEntriesWeight(nBackRaces,'fCondition',nDefaultVal)
SrEntriesWeight(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric (Width=3)

### 16.3.16 HrFavorite( ), JrFavorite( ), SrFavorite( )

The functions return a single character specifying the favorite status of a horse. Horses in a race will be divided into two groups: higher investment group and lower investment group. Horses in the higher investment (lower final odds) group will be assigned codes "1","2","3", and so on, while horses in the lower investment group (higher final odds) will be assigned codes "A", "B", "C", and so on. If the number of horses is an odd number, horses in the higher investment group will be more than that in the lower investment group.

Horses of higher investments (lower odds):

| Code | Favorite Status |
|---|---|
| 1 | Most favorite |
| 2 | Second favorite |
| 3 | Third favorite |
| … | (and so on up to half of total number of horses) |

Horses of lower investments (higher odds):

| Code | Favorite Status |
|---|---|
| A | Least favorite |
| B | Second least favorite |
| C | Third least favorite |
| … | (and so on up to half of total number of horses) |

In the example shown in the following view, the HrFavorite( ) function is applied to race 157 of season 02-03. Horses in the view are sorted by final odds.

© Welkin Systems Limited

| | Horse | Win Ticket | Final Odds | HrFavorite() |
|---|---|---|---|---|
| | Super Blitz | 4050 | 4.3 | 1 |
| | Snowstorm | 2960 | 5.9 | 2 |
| | Winwin Combo | 2580 | 6.7 | 3 |
| | Don Sebastian | 2540 | 6.9 | 4 |
| | Danriva | 2060 | 8.5 | 5 |
| | Round The World | 1910 | 9.1 | 6 |
| | High Sky | 1620 | 10.0 | F |
| | Epic Expression | 1200 | 14.0 | E |
| | Koh-i-noor | 1090 | 16.0 | D |
| | Black Knight | 590 | 29.0 | C |
| | Mascot Treasure | 380 | 45.0 | B |
| | Northern Flyer | 220 | 78.0 | A |

**Syntax**

```
HrFavorite(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrFavorite(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrFavorite(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

**See also**

HrWinTickets( ), JrWinTickets( ), SrWinTickets( ) | HrPlaceTickets( ), JrPlaceTickets ( ), SrPlaceTickets( ) | HrOdds( ), JrOdds( ), SrOdds( ) | HrFinalOdds( ), JrFinalOdds ( ), SrFinalOdds( )

### 16.3.17 HrFinalOdds( ), JrFinalOdds( ), SrFinalOdds( )

The functions return the final odds of the horse.

**Syntax**

```
HrFinalOdds(nBackRaces,'fCondition',nDefaultVal)
JrFinalOdds(nBackRaces,'fCondition',nDefaultVal)
SrFinalOdds(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**See Also**

HrOdds( )

### 16.3.18 HrFP( ), JrFP( ), SrFP( )

The functions return the finishing position of the horse.

**Syntax**

© Welkin Systems Limited

```
HrFP(nBackRaces,'fCondition',nDefaultVal)
JrFP(nBackRaces,'fCondition',nDefaultVal)
SrFP(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

### 16.3.19 HrFPX( ), JrFPX( ), SrFPX( )

These functions are identical to HrFP( ), JrFP( ) and SrFP( ) except the following.

1. When there are horses with the same finishing position, the functions will return different numbers for the horses. For example, if two horses gets position 3, the function will return 3 and 4 for these horses.
2. The return values of these functions are guaranteed unique for each starter within a race.

**Syntax**

```
HrFPX(nBackRaces,'fCondition',nDefaultVal)
JrFPX(nBackRaces,'fCondition',nDefaultVal)
SrFPX(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**See Also**

HrFP( ), JrFP( ), SrFP( )

### 16.3.20 HrHorseNo( ), JrHorseNo( ), SrHorseNo( )

The functions return the horse number of the horse.

**Syntax**

```
HrHorseNo(nBackRaces,'fCondition',nDefaultVal)
JrHorseNo(nBackRaces,'fCondition',nDefaultVal)
SrHorseNo(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

### 16.3.21 HrHorseNoStr( ), JrHorseNoStr( ), SrHorseNoStr( )

This function is similar to the HrHorseNo( ) function set except that the functions return the horse number in a character string. If there is no horse number for the horse, a blank string, composed of a character of ASCII 254 and a space, will be returned. When you sort on the column in ascending order, the first character blank string (ASCII 254) will make the line sink to the bottom of the table.

© Welkin Systems Limited

**Syntax**

```
HrHorseNoStr(nBackRaces,'fCondition',nDefaultVal)
JrHorseNoStr(nBackRaces,'fCondition',nDefaultVal)
SrHorseNoStr(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Charcter (Width=2)

## 16.3.22 HrIsRunner( ), JrIsRunner( ), SrIsRunner( )

If the horse does not run in the race, these functions return a True value. If the horse is in the entries list but dose not start, False is returned.

**Syntax**

```
HrIsRunner( )
JrIsRunner( )
SrIsRunner( )
```

**Returns**

Logical

**See also**

HrRaceRunning( ), JrRaceRunning( ), SrRaceRunning( )

## 16.3.23 HrLastSectPos( ), JrLastSectPos( ), SrLastSectPos( )

The functions return the running positions of the horse in a particular section.

**Syntax**

```
HrLastSectPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrLastSectPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrLastSectPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

If you omit (or pass 1 into) the first parameter, the function will return the position of the last section.

**Returns**

Numeric

**See Also**

HrSectionPos( ), HrPositions( )

© Welkin Systems Limited

### 16.3.24 HrLoadDiff( ), JrLoadDiff( ), SrLoadDiff( )

The functions return the difference in weight (before deducting apprentice allowance) of the horse between the underlying race and previous race.

#### Syntax

```
HrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
JrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
SrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

#### See Also

HrWeight( ), HrLoadDiffStr( ), HrNetLoadDiff( )

### 16.3.25 HrLoadDiffStr( ), JrLoadDiffStr( ), SrLoadDiffStr( )

The functions are similar to the HrLoadDiff( ), JrLoadDiff( ), and SrLoadDiff( ) functions except that the loading difference is returned in a character string with a '+' sign in positive value and '-' sign in negative value.

#### Syntax

```
HrLoadDiffStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrLoadDiffStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrLoadDiffStr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=3)

#### See Also

HrWeight( ), HrLoadDiff( ), HrNetLoadDiff( )

### 16.3.26 HrMargin( ), JrMargin( ), SrMargin( )

The functions return the margin between the winning horse and the underlying horse.

#### Syntax

```
HrMargin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrMargin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrMargin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=10)

© Welkin Systems Limited

**See also**

HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( )

## 16.3.27 HrMarginLen( ), JrMarginLen( ), SrMarginLen( )

The functions return the numeric margin value between the winning horse and the underlying horse according to the following conversion rules:

| Margin | Value |
|---|---|
| Nose | 0.01 |
| Short Head | 0.05 |
| Head | 0.10 |
| Short Neck | 0.20 |
| Neck | 0.25 |
| Multi-length or not reaching destination | 99.00 |

**Syntax**

```
HrMarginLen(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrMarginLen(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrMarginLen(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (Width=5, Dec=2)

**See also**

HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( ) | HrMargin( ), JrMargin( ), SrMargin( )

## 16.3.28 HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( )

These functions are similar to the functions HrMarginLen( ), JrMarginLen( ), and SrMarginLen( ) except that when the function is applied to the winner, the distance (which will be made negative) between the winner and the runner-up is returned.

**Syntax**

```
HrMarginLenX(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrMarginLenX(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrMarginLenX(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (Max=99.00, Width=5, Dec=2)

**See also**

HrMarginLen( ), JrMarginLen( ), SrMarginLen( ) | HrMargin( ), JrMargin( ), SrMargin( )

© Welkin Systems Limited

## 16.3.29 HrNetLoadDiff( ), JrNetLoadDiff( ), SrNetLoadDiff( )

The functions return the difference in weight (after deducting apprentice allowance) of the horse between the underlying race and previous race.

### Syntax

```
HrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
JrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
SrLoadDiff(nBackRaces,'fCondition',nDefaultVal)
```

### Returns

Numeric

### See Also

HrWeight( ), HrLoadDiffStr( ), HrLoadDiff( )

## 16.3.30 HrOdds( ), JrOdds( ), SrOdds( )

The HrOdds( ) function returns the a string showing the odds in the following format: "5.6 6.1 6.7"

In the string, 5.6 is the overnight odds; 6.1 is the odds 5 minutes before the race; 6.7 is the final odds.

### Syntax

```
HrOdds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrOdds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOdds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=13)

### See Also

ListVal( ), ListItem( ), RListVal( ), RListItem( )

## 16.3.31 HrOdds1Day( ), JrOdds1Day( ), SrOdds1Day( )

Return the overnight odds of the horse. Before 6/3/2009, the collection time 00:00 am on the racing day. Since 7/3/2009, it has be changed to 8:00 pm one day prior to the racing day.

### Syntax

```
HrOdds1Day(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrOdds1Day(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOdds1Day(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

© Welkin Systems Limited

Numeric (Width=4, Decimal=1)

### 16.3.32 HrOdds5Min( ), JrOdds5Min( ), SrOdds5Min( )

For races before 7/3/2009 the functions return the odds of the horse 5 minutes before the race. For races on or after 7/3/2009 the functions return the odds of the horse 60 minutes before the race.

#### Syntax

```
HrOdds5Min(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrOdds5Min(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOdds5Min(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric (Width=4, Decimal=1)

### 16.3.33 HrOverweight( ), JrOverweight( ), SrOverweight( )

The functions return the overweight figure of a horse in a start.

#### Syntax

```
HrOverweight(nBackRaces,'fCondition',nDefaultVal)
JrOverweight(nBackRaces,'fCondition',nDefaultVal)
SrOverweight(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric (width=2)

#### See Also

HrWeight( ), JrWeight( ), SrWeight( ); HrAllowance( ), JrAllowance( ), SrAllowance( )

### 16.3.34 HrPastIncident( ), JrPastIncident( ), HrPastIncident( )

The functions return the Jockey Club past incident extract

#### Syntax

```
HrPastIncident(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPastIncident(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPastIncident(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width equals number of characters of the incident extract)

© Welkin Systems Limited

### 16.3.35 HrPlaceTickets( ), JrPlaceTickets( ), SrPlaceTickets( )

The functions return the place investment for the horse in unit of thousand.

#### Syntax

```
HrPlaceTickets(nBackRaces,'fCondition',nDefaultVal)
JrPlaceTickets(nBackRaces,'fCondition',nDefaultVal)
SrPlaceTickets(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

#### See Also

HrWinTickets( )

### 16.3.36 HrPositions( ), JrPositions( ), SrPositions( )

The functions return a string showing the running positions of the horse in the format: "10 12 11"

#### Syntax

```
HrPositions(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPositions(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPositions(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=25)

#### See Also

ListVal( ), ListItem( ), RListVal( ), RListItem( )

### 16.3.37 HrPrivilege( ), JrPrivilege( ), SrPrivilege( )

Special start priority in entries declared by the stable. The functions will return one of the following values:

- `"Trump Card"`
- `"Priority  "`
- `"Priority+Trump"`
- `"          "` (blank string)

#### Syntax

```
HrPrivilege(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPrivilege(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPrivilege(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

© Welkin Systems Limited

Character (Width=14 in English Mode, 9 in Chinese Mode)

**See also**

HrPrivilegeID( ), JrPrivilegeID( ), SrPrivilegeID( ) |

## 16.3.38 HrPrivilegeID( ), JrPrivilegeID( ), SrPrivilegeID( )

RaceMate internal ID of the special start priority of a horse in entries declared by the stable. The IDs are as follows:

| ID | Meaning |
|---|---|
| E | Horse with priority to run |
| T | Trump card |
| B | Priority+Trump |
| (Blank) | No special priority |

**Syntax**

```
HrPrivilegeID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPrivilegeID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPrivilegeID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

## 16.3.39 HrRunning( ), JrRunning( ), SrRunning( )

The functions return the Jockey Club horse running comments

**Syntax**

```
HrRaceRunning(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceRunning(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceRunning(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width equals number of characters of the comment)

## 16.3.40 HrRacingWeight( ), JrRacingWeight( ), SrRacingWeight( )

The functions return the body weight of the horse before races start.

**Syntax**

```
HrRacingWeight(nBackRaces,'fCondition',nDefaultVal)
JrRacingWeight(nBackRaces,'fCondition',nDefaultVal)
SrRacingWeight(nBackRaces,'fCondition',nDefaultVal)
```

© Welkin Systems Limited

**Returns**

Numeric

### 16.3.41 HrRating( ), JrRating( ), SrRating( )

The functions return the Jockey Club rating of the horse.

**Syntax**

```
HrRating(nBackRaces,'fCondition',nDefaultVal)
JrRating(nBackRaces,'fCondition',nDefaultVal)
SrRating(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**See Also**

HrCurRating( ), HrRaceRating( )

### 16.3.42 HrRatingDiff( ), JrRatingDiff( ), SrRatingDiff( )

The functions return the change of horse rating in underlying race comparing with last race. A positive value is for an increase while a negative value is for an decrease.

**Syntax**

```
HrRatingDiff(nBackRaces,'fCondition',nDefaultVal)
JrRatingDiff (nBackRaces,'fCondition',nDefaultVal)
SrRatingDiff (nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

### 16.3.43 HrReserved( ), JrReserved( ), SrReserved( )

The functions return the reserve number, if any, of the horse.

**Syntax**

```
HrReserverd(nBackRaces,'fCondition',nDefaultVal)
JrReserverd(nBackRaces,'fCondition',nDefaultVal)
SrReserverd(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

© Welkin Systems Limited

## 16.3.44 HrSeconds( ), JrSeconds( ), SrSeconds( )

The functions return the time in seconds that the horse takes to finish a running.

### Syntax

```
HrSeconds(nBackRaces,'fCondition',nDefaultVal)
JrSeconds(nBackRaces,'fCondition',nDefaultVal)
SrSeconds(nBackRaces,'fCondition',nDefaultVal)
```

### Returns

Numeric

### See Also

HrSpeed( ), HrTime( )

## 16.3.45 HrSectionPos( ), JrSectionPos( ), SrSectionPos( )

The functions return the running positions of the horse in a particular section.

### Syntax

```
HrSectionPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectionPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectionPos(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Numeric

### See Also

HrLastSectPos( ), HrPositions( )

## 16.3.46 HrSectMargins( ), JrSectMargins( ), SrSectMargins( )

If the 1st argument is omitted, the functions return the margins behind the fastest horse of all sections in a character string. If a number if passed into the 1st argument, the margins behind the fastest horse of the section will be returned.

### Syntax

```
HrSectMargins(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectMargins(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectMargins(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Arguments

- If nSection is omitted or equals to 0, margin behind the leading horse of all sections will be returned in a character string.
- When nSection is a positive number, it is a section number counting from the starting point (i.e., 1 for the first section and 2 for the second, etc.). The margin behind the leading horse for the section will be returned as a character sting.

© Welkin Systems Limited

- When nSection is a negative number, it is a section number counting from the finishing point (i.e., 1 for the last section and 2 for the 2nd last, etc.). The margin behind the leading horse for the section will be returned as a character sting.

See Arguments of MCL functions for details about other arguments.

**Returns**

Character (width=10)

**See Also**

HrSectPositions( ), JrSectPositions( ), SrSectPositions( ) | HrSectTime( ), JrSectTime ( ), SrSectTime( ) | HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( ) | HrTime( ), JrTime( ), SrTime( ) | HrSeconds( ), JrSeconds( ), SrSeconds( ) | HrMargin ( ), JrMargin( ), SrMargin( ) | HrMarginLen( ), JrMarginLen( ), SrMarginLen( ) | HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( )

## 16.3.47 HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( )

If the 1st argument is omitted, the functions return the margins behind the fastest horse of all sections in a character string composed of a list of decimal numbers. If a number if passed into the 1st argument, the margins behind the fastest horse of the section will be returned in numeric values.

**Syntax**

```
HrSectMargLens(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectMargLens(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectMargLens(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Arguments**

- If nSection is omitted or equals to 0, margin behind the leading horse of all sections will be returned in a character string composed of a list of decimal numbers.
- When nSection is a positive number, it is a section number counting from the starting point (i.e., 1 for the first section and 2 for the second, etc.). The margin behind the leading horse for the section will be returned as a numeric value.
- When nSection is a negative number, it is a section number counting from the finishing point (i.e., 1 for the last section and 2 for the 2nd last, etc.). The margin behind the leading horse for the section will be returned as a numeric value.

See Arguments of MCL functions for details about other arguments.

**Returns**

Numeric (decimal=2)

**See Also**

HrSectPositions( ), JrSectPositions( ), SrSectPositions( ) | HrSectTime( ), JrSectTime ( ), SrSectTime( ) | HrTime( ), JrTime( ), SrTime( ) | HrSeconds( ), JrSeconds( ), SrSeconds( ) | HrMargin( ), JrMargin( ), SrMargin( ) | HrMarginLen( ), JrMarginLen( ), SrMarginLen( ) | HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( ) | HrSectMargins ( ), JrSectMargins( ), SrSectMargins( )

© Welkin Systems Limited

## 16.3.48 HrSectMargLensX( ), JrSectMargLensX( ), SrSectMargLensX( )

If the 1st argument is omitted, the functions return the margins behind the fastest horse of all sections in a character string composed of a list of decimal numbers (identical to HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( )). If a number if passed into the 1st argument, the margins behind the fastest horse of the section will be returned in numeric values. However, for the fastest horse, the length which leads the second horse will be returned in a negative number.

### Syntax

```
HrSectMargLensx(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectMargLensx(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectMargLensx(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Arguments

- If nSection is omitted or equals to 0, margin behind the leading horse of all sections will be returned in a character string composed of a list of decimal numbers.
- When nSection is a positive number, it is a section number counting from the starting point (i.e., 1 for the first section and 2 for the second, etc.). The margin behind the leading horse for the section will be returned as a numeric value.
- When nSection is a negative number, it is a section number counting from the finishing point (i.e., 1 for the last section and 2 for the 2nd last, etc.). The margin behind the leading horse for the section will be returned as a numeric value.

See Arguments of MCL functions for details about other arguments.

### Returns

Numeric (decimal=2)

### See Also

HrSectPositions( ), JrSectPositions( ), SrSectPositions( ) | HrSectTime( ), JrSectTime ( ), SrSectTime( ) | HrTime( ), JrTime( ), SrTime( ) | HrSeconds( ), JrSeconds( ), SrSeconds( ) | HrMargin( ), JrMargin( ), SrMargin( ) | HrMarginLen( ), JrMarginLen( ), SrMarginLen( ) | HrMarginLenX( ), JrMarginLenX( ), SrMarginLenX( ) | HrSectMargins ( ), JrSectMargins( ), SrSectMargins( )

## 16.3.49 HrSectPositions( ), JrSectPositions( ), SrSectPositions( )

If the 1st argument is omitted, the functions return the positions of all sections of the underlying horse. If a number if passed into the 1st argument, the position of a particular section will be returned.

### Syntax

```
HrSectPositions(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectPositions(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectPositions(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Arguments

- If nSection is omitted or equals to 0, positions of all sections will be returned in a character string.
- When nSection is a positive number, it is a section number counting from the

© Welkin Systems Limited

starting point (i.e., 1 for the first section and 2 for the second, etc.). The position for the section will be returned as a number.
- When nSection is a negative number, it is a section number counting from the finishing point (i.e., 1 for the last section and 2 for the 2nd last, etc.). The position for the section will be returned as a number.

See Arguments of MCL functions for details about other arguments.

#### Returns

Character if nSection argument is ignored. Numeric if nSection argument if used

#### See Also

HrSectTime( ), JrSectTime( ), SrSectTime( ) | HrSectMargins( ), JrSectMargins( ), SrSectMargins( ) | HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( ) | HrTime ( ), JrTime( ), SrTime( ) | HrSeconds( ), JrSeconds( ), SrSeconds( )

### 16.3.50 HrSectTime( ), JrSectTime( ), SrSectTime( )

If the 1st argument is omitted, the functions return the time of all sections of the underlying horse. If a number if passed into the 1st argument, the time of a particular section will be returned.

#### Syntax

```
HrSectTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Arguments

- If nSection is omitted or equals to 0, times of all sections will be returned in a character string.
- When nSection is a positive number, it is a section number counting from the starting point (i.e., 1 for the first section and 2 for the second, etc.). The time for the section will be returned as a number.
- When nSection is a negative number, it is a section number counting from the finishing point (i.e., 1 for the last section and 2 for the 2nd last, etc.). The time for the section will be returned as a number.

See Arguments of MCL functions for details about other arguments.

#### Returns

Character (width=34) if nSection argument is ignored. Numeric if nSection argument if used

#### See Also

HrSectMargins( ), JrSectMargins( ), SrSectMargins( ) | HrSectPositions( ), JrSectPositions( ), SrSectPositions( ) | HrSectMargLens( ), JrSectMargLens( ), SrSectMargLens( ) | HrTime( ), JrTime( ), SrTime( ) | HrSeconds( ), JrSeconds( ), SrSeconds( )

© Welkin Systems Limited

### 16.3.51 HrSectTimew( ), JrSectTimew( ), SrSectTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrSectTime( ), JrSectTime( ), SrSectTime( )

#### Returns

Character (width=40) if nSection argument is ignored. Numeric if nSection argument if used

### 16.3.52 HrSpecialBit( ), JrSpecialBit( ), SrSpecialBit( )

The functions return special bit and gears for the horse.

#### Syntax

```
HrSpecialBit(nBackRaces,'fCondition',cDefaultVal)
JrSpecialBit(nBackRaces,'fCondition',cDefaultVal)
SrSpecialBit(nBackRaces,'fCondition',cDefaultVal)
```

#### Returns

Character (Chars/Digits=109 in English mode, 74 in Chinese mode)

#### See Also

HrBlinker( ), HrBlinkerAbbr( ), HrBlinkerAbbrNew( ), HrSpecialBitAbbr( )

### 16.3.53 HrSpecialBitAbbr( ), JrSpecialBitAbbr( ), SrSpecialBitAbbr( )

The functions return abbreviations for the special bit and gears for the horse.

#### Syntax

```
HrSpecialBitAbbr(nBackRaces,'fCondition',cDefaultVal)
JrSpecialBitAbbr(nBackRaces,'fCondition',cDefaultVal)
SrSpecialBitAbbr(nBackRaces,'fCondition',cDefaultVal)
```

#### Returns

Character (Chars/Digits=16)

#### See Also

HrBlinker( ), HrBlinkerAbbr( ), HrBlinkerAbbrNew( ), HrSpecialBit( )

© Welkin Systems Limited

## 16.3.54 HrSpeed( ), JrSpeed( ), SrSpeed( )

The functions return the speed in meters per second of the horse.

### Syntax

```
HrSpeed(nBackRaces,'fCondition',nDefaultVal)
JrSpeed(nBackRaces,'fCondition',nDefaultVal)
SrSpeed(nBackRaces,'fCondition',nDefaultVal)
```

### Returns

Numeric

### See Also

HrSeconds( ), HrTime( )

## 16.3.55 HrStablePrefence( ), JrStablePrefence( ), SrStablePrefence( )

Order of Trainer Preference represented by a number in Entries List.

### Syntax

```
HrStablePreference(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStablePreference(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStablePreference(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Numeric (Width=2)

### Example of Returned Value

2

### See also

HrPrivilege( ), JrPrivilege( ), SrPrivilege( )

## 16.3.56 HrStartID( ), JrStartID( ), SrStartID( )

The functions return a unique character string representing a particular start of the horse.

### Syntax

```
HrStartID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStartID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStartID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=8)

© Welkin Systems Limited

## 16.3.57 HrStartIDOn(), JrStartIDOn(), SrStartIDOn()

The functions return the "Start ID" of a horse (jockey, stable) run on specified date. If no run is found on the day or "B" is passed into the 2nd argument, the Start ID of the suitable run on a nearest day will be returned. If there is more than one run on the day, the Start ID of the day's last run will be returned.

### Syntax

```
HrStartIDOn(dDate,cFlag)
JrStartIDOn(dDate,cFlag)
SrStartIDOn(dDate,cFlag)
```

### Arguments

dDate

Target date. The Start ID of an eligible start on this day will be returned. If no eligible start is found on this date, the Start ID of an eligible start on the nearest day after (cFlag="A") or before (cFlag="B") this date will be returned.

cFlag

"A" or omitted: The Start ID of an start on dDate will be returned. If no eligible start is found on dDate, the ID of an eligible start on the nearest day after this date will be returned. If there is more than one eligible start on the day, the Start ID of the day's last start will be returned.

"B": The Start ID of an eligible start before dDate will be returned. If there is more than one eligible start on the day, the Start ID of the day's last start will be returned.

### Returns

Character (Width=8)

### See Also

HrStartID( ), JrStartID( ), SrStartID( ) || HrRaceID( ), JrRaceID( ), SrRaceID( )

## 16.3.58 HrStakes( ), JrStakes( ), SrStakes( )

The functions return the accumulated stakes of the horse. Please note that the functions always return the total stakes up to the season the underlying start takes place rather than the stakes up to a particular race. Therefore you may see the same figure returned by these functions on different races of the same season.

### Warning

Stakes data are stored season by season. These functions will give accurate figures only for new races or the last starts of horses of a season.

### Syntax

```
HrStakes(cFlag)
```

© Welkin Systems Limited

```
JrStakes(cFlag)
SrStakes(cFlag)
```

| cFlag | For |
|---|---|
| Omitted or blank | Accumulated stakes up to the season of the underlying start |
| "A" | Average accumulated stakes per race |
| "L" | Local stakes |
| "O" | Overseas stakes |
| An Arabic numeral for number of seasons | Number of latest seasons in which stakes earned |

**About Season Number**

Once a season number is included into the cFlag argument, the stakes the horse earned before arriving HK will not be counted.

**Returns**

Numeric (9 digits)

**Examples**

Total local and overseas stakes earned in the current season which the start takes place.

```
HrStakes("1")
```

Total overseas stakes earned in all seasons up to current start. Stakes earned before arriving HK will also be included.

```
HrStakes("O")
```

Total local stakes earned in all seasons up to current start.

```
HrStakes("L")
```

Local stakes earned in current season.

```
HrStakes("L1")
```

Local stakes earned in latest 2 seasons.

```
HrStakes("L2")
```

Total Local and overseas stakes earned in latest 2 seasons. Stakes earned before arriving HK **will not** be included.

```
HrStakes("2LO")
```

Average stakes per race earned in latest 2 seasons. Stakes earned before coming to HK will not be counted.

```
HrStakes("2A")
```

Average stakes per race, including those earned before coming to HK.

```
HrStakes("A")
```

© Welkin Systems Limited

## 16.3.59 HrTime( ), JrTime( ), SrTime( )

The functions return the time taken by the horse to finish the running. Returned values will be in the following format:

1.23.9 (1 minute and 23.9 seconds)

### Syntax

```
HrTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=6)

### See Also

HrSeconds( ), HrSpeed( )

## 16.3.60 HrTimew( ), JrTimew( ), SrTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrTime( ), JrTime( ), SrTime( )

### Returns

Character (Width=7)

## 16.3.61 HrUpdown( ), JrUpdown( ), SrUpdown( )

The function returns the information about the change of the class of the horse. A list of possible returned value is shown below.

| English | Chinese |
|---|---|
| Up | 升班 |
| Down | 降班 |
| Recls | 插班 |
| V.Promote | 越級 |
| V.Demote | 降級 |

### Syntax

```
HrUpdown(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrUpdown(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrUpdown(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

© Welkin Systems Limited

Character (Width=10)

**See Also**

HrUpdownID( )

## 16.3.62 HrUpdownID( ), JrUpdownID( ), SrUpdownID( )

The function returns an ID that identifies the change of the class of the horse. The following list shows the IDs. Space will be returned if no change.

| ID | For |
|---|---|
| U | Up |
| D | Down |
| I | Recls |
| J | V.Promote |
| L | V.Demote |

**Syntax**

```
HrUpdownID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrUpdownID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrUpdownID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

**See Also**

HrUpdown( )

## 16.3.63 HrWeight( ), JrWeight( ), SrWeight( )

The functions return the net loading (apprentice allowance already deducted) of the horse.

**Syntax**

```
HrWeight(nBackRaces,'fCondition',nDefaultVal)
JrWeight(nBackRaces,'fCondition',nDefaultVal)
SrWeight(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

© Welkin Systems Limited

### 16.3.64 HrWinTickets( ), JrWinTickets( ), SrWinTickets( )

The functions return the win investment for the horse in unit of thousand.

**Syntax**

```
HrWinTickets(nBackRaces,'fCondition',nDefaultVal)
JrWinTickets(nBackRaces,'fCondition',nDefaultVal)
SrWinTickets(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**See Also**

HrPlaceTickets( )

### 16.3.65 HrWQPStat( ), JrWQPStat( ), SrWQPStat( )

The functions return in a string the numbers of Win, Quinella, Place, and unplaced in the horse's racing record.

**Syntax**

```
HrWQPStat(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrWQPStat(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrWQPStat(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=26)

**Example**

"0, 3, 0,19 ( 22) "

## 16.4 Static information of Horse, Jockey, and Stable

This group of functions returns particulars about horses, jockeys, and stables, which will not change in different races.

**>>> Subtopics of this section <<<**

- HrArrivalYear( ), JrArrivalYear( ), SrArrivalYear( )
- HrBrandNo( ), JrBrandNo( ), SrBrandNo( )
- HrBreedNote( ), JrBreedNote( ), SrBreedNote( )
- HrCHorse( ), JrCHorse( ), SrCHorse( )
- HrColor( ), JrColor( ), SrColor( )
- HrCountryID( ), JrCountryID( ), SrCountryID( )
- HrDam( ), JrDam( ), SrDam( )
- HrDDam( ), JrDDam( ), SrDDam( )
- HrDDDam( ), JrDDDam( ), SrDDDam( )

© Welkin Systems Limited

- HrDDSire( ), JrDDSire( ), SrDDSire( )
- HrDSDam( ), JrDSDam( ), SrDSDam( )
- HrDSire( ), JrDSire( ), SrDSire( )
- HrDSSire( ), HrDSSire( ), SrDSSire( )
- HrEHorse( ), JrEHorse( ), SrEHorse( )
- HrFoalDate( ), JrFoalDate( ), SrFoalDate( )
- HrHorse( ), JrHorse( ), SrHorse( )
- HrHorseID( ), JrHorseID( ), SrHorseID( )
- HrHorseMarkNote( ), JrHorseMarkNote( ), SrHorseMarkNote( )
- HrHorseNote( ), JrHorseNote( ), SrHorseNote( )
- HrJockey( ), JrJockey( ), SrJockey( )
- HrJockeyClass( ), JrJockeyClass( ), SrJockeyClass( )
- HrJockeyEngAbbr( ), JrJockeyEngAbbr( ), SrJockeyEngAbbr( )
- HrJockeyID( ), JrJockeyID( ), SrJockeyID( )
- HrJockeyRetired( ), JrJockeyRetired( ), SrJockeyRetired( )
- HrNoteMark( ), JrNoteMark( ), SrNoteMark( )
- HrOrigin( ), JrOrigin( ), SrOrigin( )
- HrOriginCategory( ), JrOriginCategory( ), SrOriginCategory( )
- HrOwner( ), JrOwner( ), SrOwner( )
- HrRetired( ), JrRetired( ), SrRetired( )
- HrSameBlood( ), JrSameBlood( ), SrSameBlood( )
- HrSDam( ), JrSDam( ), SrSDam( )
- HrSDDam( ), JrSDDam( ), SrSDDam( )
- HrSDSire( ), JrSDSire( ), SrSDSire( )
- HrSex( ), JrSex( ), SrSex( )
- HrSickHistory( ), JrSickHistory( ), SrSickHistory( )
- HrSire( ), JrSire( ), SrSire( )
- HrSSDam( ), JrSSDam( ), SrSSDam( )
- HrSSire( ), JrSSire( ), SrSSire( )
- HrSSSire( ), JrSSSire( ), SrSSSire( )
- HrStable( ), JrStable( ), SrStable( )
- HrStableAbbr( ), JrStableAbbr( ), SrStableAbbr( )
- HrStableID( ), JrStableID( ), SrStableID( )
- HrStableRetired( ), JrStableRetired( ), SrStableRetired( )

### 16.4.1 HrArrivalYear( ), JrArrivalYear( ), SrArrivalYear( )

Year of arrival of the horse

#### Syntax

```
HrArrivalYear( )
JrArrivalYear(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrArrivalYear(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric (Width=4)

#### See also

HrOrigin( ), JrOrigin( ), SrOrigin( ) | HrOriginCategory( ), JrOriginCategory( ), SrOriginCategory( ) | HrCountryID( ), JrCountryID( ), SrCountryID( ) | HrAge( ), JrAge ( ), SrAge( )

© Welkin Systems Limited

### 16.4.2 HrBrandNo( ), JrBrandNo( ), SrBrandNo( )

The functions return the brand number of the horse.

**Syntax**

```
HrBrandNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrBrandNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBrandNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=4)

### 16.4.3 HrBreedNote( ), JrBreedNote( ), SrBreedNote( )

The functions return the user comments on the Pedigree page of the Horse Form.

**Syntax**

```
HrBreedNote()
JrBreedNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrBreedNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width equals number of characters of the user comment)

**See Also**

HrNoteMark( ), HrHorseMarkNote( ), HrHorseNote( )

### 16.4.4 HrCHorse( ), JrCHorse( ), SrCHorse( )

The functions return the Chinese name of the horse.

**Syntax**

```
HrCHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrCHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=8)

**See Also**

HrHorse( ), HrEHorse( ) | HrEHorse( ), JrHorse( ), SrHorse( )

© Welkin Systems Limited

### 16.4.5 HrColor( ), JrColor( ), SrColor( )

Colour of Horse

#### Syntax

```
HrColor( )
JrColor(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrColor(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=4 in Chinese mode, Width=13 in English mode)

### 16.4.6 HrCountryID( ), JrCountryID( ), SrCountryID( )

ID code for the origin country of the horse.

#### Syntax

```
HrCountryID( )
JrCountryID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCountryID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=3)

#### See also

HrOrigin( ), JrOrigin( ), SrOrigin( ) | HrOriginCategory( ), JrOriginCategory( ), SrOriginCategory( )

### 16.4.7 HrDam( ), JrDam( ), SrDam( )

The functions return the name of the horse's dam.

#### Syntax

```
HrDam( )
JrDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=22)

#### See Also

HrSire( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

© Welkin Systems Limited

### 16.4.8 HrDDam( ), JrDDam( ), SrDDam( )

The functions return the name of the horse's dam's dam.

#### Syntax

```
HrDDam( )
JrDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=22)

#### See Also

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

### 16.4.9 HrDDDam( ), JrDDDam( ), SrDDDam( )

The functions return the name of the horse's dam's dam's dam.

#### Syntax

```
HrDDDam( )
JrDDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=22)

#### See Also

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam ( ), HrSDDam( )

### 16.4.10 HrDDSire( ), JrDDSire( ), SrDDSire( )

The functions return the name of the horse's dam's sire's sire.

#### Syntax

```
HrDDSire( )
JrDDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=22)

© Welkin Systems Limited

**See also**

HrSire( ), JrSire( ), SrSire( ) | HrDam( ), JrDam( ), SrDam( ) | HrSSire( ), JrSSire( ), SrSSire( ) | HrDSire( ), JrDSire( ), SrDSire( ) | HrSDam( ), JrSDam( ), SrSDam( ) | HrDDam( ), JrDDam( ), SrDDam( ) | HrSSSire( ), JrSSSire( ), SrSSSire( ) | HrDSSire( ), HrDSSire( ), SrDSSire( ) | HrSDSire( ), JrSDSire( ), SrSDSire( ) | HrSSDam( ), JrSSDam ( ), SrSSDam( ) | HrDSDam( ), JrDSDam( ), SrDSDam( ) | HrSDDam( ), JrSDDam( ), SrSDDam( ) | HrDDDam( ), JrDDDam( ), SrDDDam( )

## 16.4.11 HrDSDam( ), JrDSDam( ), SrDSDam( )

The functions return the name of the horse's dam's sire's dam.

**Syntax**

```
HrDSDam( )
JrDSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=22)

**See Also**

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrSDDam ( ), HrDDDam( )

## 16.4.12 HrDSire( ), JrDSire( ), SrDSire( )

The functions return the name of the horse's dam's sire.

**Syntax**

```
HrDSire( )
JrDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=22)

**See Also**

HrSire( ), HrDam( ), HrSSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

© Welkin Systems Limited

## 16.4.13 HrDSSire( ), HrDSSire( ), SrDSSire( )

The functions return the name of the horse's dam's sire's sire.

### Syntax

```
HrDSSire( )
JrDSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=22)

### See Also

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam( ), HrDDDam( )

## 16.4.14 HrEHorse( ), JrEHorse( ), SrEHorse( )

The functions return the English name of the horse.

### Syntax

```
HrEHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrEHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrEHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### See Also

HrHorse( ), HrCHorse( )

## 16.4.15 HrFoalDate( ), JrFoalDate( ), SrFoalDate( )

The functions return the foaling date of the horse.

### Syntax

```
HrFoalDate(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrFoalDate(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrFoalDate(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Date

### See Also

HrAge( )

© Welkin Systems Limited

### 16.4.16 HrHorse( ), JrHorse( ), SrHorse( )

The functions return the name of the horse. In Chinese language mode, Chinese horse name will be returned. In English language mode, English horse name will be returned.

**Syntax**

```
HrHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrHorse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=8 in Chinese Mode, Width=30 in English Mode)

**See Also**

HrCHorse( ), HrEHorse( )

### 16.4.17 HrHorseID( ), JrHorseID( ), SrHorseID( )

The functions return a unique character string identifying the horse.

**Syntax**

```
HrHorseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrHorseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrHorseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=3)

### 16.4.18 HrHorseMarkNote( ), JrHorseMarkNote( ), SrHorseMarkNote( )

The functions return the user mark note of a horse inputted in the Horse Form.

**Syntax**

```
HrHorseMarkNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrHorseMarkNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrHorseMarkNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width equals number of characters of the user comment)

**See Also**

HrNoteMark( ), HrHorseNote( )

© Welkin Systems Limited

## 16.4.19 HrHorseNote( ), JrHorseNote( ), SrHorseNote( )

The functions return the user comments of a horse inputted in the Horse Form.

### Syntax

```
HrHorseNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrHorseNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrHorseNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width equals number of characters of the user comment)

### See Also

HrNoteMark( ),HrHorseMarkNote( )

## 16.4.20 HrJockey( ), JrJockey( ), SrJockey( )

The functions return the name of the jockey riding the horse. In Chinese language mode, Chinese horse name will be returned. In English language mode, English horse name will be returned.

### Syntax

```
HrJockey(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrJockey(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrJockey(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=8 in Chinese Mode, Width=30 in English Mode)

## 16.4.21 HrJockeyClass( ), JrJockeyClass( ), SrJockeyClass( )

Category of the horse. The function will return one of the following values:

- Foreign
- Local
- Apprentice

### Syntax

```
HrJockeyClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrJockeyClass( )
SrJockeyClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=10 in English mode, Width=6 in Chinese modei^

© Welkin Systems Limited

## 16.4.22 HrJockeyEngAbbr( ), JrJockeyEngAbbr( ), SrJockeyEngAbbr( )

The functions return the Jockey Club abbreviation of the jockey riding the horse. No matter in Chinese or English language modes, an abbreviation of less than 3 letters will be returned.

Special Abbreviations

| Abbreviation | For |
|---|---|
| W* | Withdrawal |
| A* | Assistant |
| AD* | Aide |
| AT* | Assistant Trainer |

### Syntax

```
HrJockeyEngAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrJockeyEngAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrJockeyEngAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=3)

## 16.4.23 HrJockeyID( ), JrJockeyID( ), SrJockeyID( )

The functions return a unique character string identifying the jockey.

### Syntax

```
HrHorseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrHorseID( )
SrHorseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=2)

## 16.4.24 HrJockeyRetired( ), JrJockeyRetired( ), SrJockeyRetired( )

Return whether the jockey is currently (not at his races) retired.

### Syntax

```
HrJockeyRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrJockeyRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrJockeyRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Logical

### See Also

© Welkin Systems Limited

HrRetired( ), JrRetired( ), SrRetired( ); HrJockey( ), JrJockey( ), SrJockey( ); HrStableRetired( ), JrStableRetired( ), SrStableRetired( )

### 16.4.25 HrNoteMark( ), JrNoteMark( ), SrNoteMark( )

The functions return a "*" if "Mark Notes" was entered in the Horse Form.

#### Syntax

```
HrNoteMark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrNoteMark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrNoteMark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=1)

#### See Also

HrHorseNote( ),HrHorseMarkNote( )

### 16.4.26 HrOrigin( ), JrOrigin( ), SrOrigin( )

Country where the horse came from.

#### Syntax

```
HrOrigin( )
JrOrigin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOrigin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=10)

#### See Also

HrCountryID( ), JrCountryID( ), SrCountryID( ) | HrOriginCategory( ), JrOriginCategory ( ), SrOriginCategory( )

### 16.4.27 HrOriginCategory( ), JrOriginCategory( ), SrOriginCategory( )

Catergory of the origin of the horse

#### Syntax

```
HrOriginCategory( )
JrOriginCategory(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOriginCategory(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=8)

© Welkin Systems Limited

**See also**

HrOrigin( ), JrOrigin( ), SrOrigin( ) | HrCountryID( ), JrCountryID( ), SrCountryID( )

## 16.4.28 HrOwner( ), JrOwner( ), SrOwner( )

Owner of the horse

**Syntax**

```
HrOwner( )
JrOwner(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrOwner(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=24)

## 16.4.29 HrRetired( ), JrRetired( ), SrRetired( )

Return whether the horse is currently (not at its starts) retired.

**Syntax**

```
HrRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Logical

**See Also**

HrJockeyRetired( ), JrJockeyRetired( ), SrJockeyRetired( ); HrHorse( ), JrHorse( ), SrHorse( ); HrStableRetired( ), JrStableRetired( ), SrStableRetired( )

## 16.4.30 HrSameBlood( ), JrSameBlood( ), SrSameBlood( )

A list of horse which are of the same blood. You may use the first argument (cGeneration) of the function to specify a generation. If the first argument is omitted, names of the horses with the same sire will be returned.

| 1st Argument | Generation |
|---|---|
| S | Same Sire |
| D | Same Dam |
| SS | Same Sire of Sire |
| DS | Same Dam of Sire |
| SD | Same Sire of Dam |
| DD | Same Dam of Dam |
| SSS | Same Sire of Sire of Sire |
| DSS | Same Dam of Sire of Sire |

© Welkin Systems Limited

| | |
|---|---|
| SDS | Same Sire of Dam of Sire |
| DDS | Same Dam of Dam of Sire |
| SSD | Same Sire of Sire of Dam |
| DSD | Same Dam of Sire of Dam |
| SDD | Same Sire of Dam of Dam |
| DDD | Same Dam of Dam of Dam |

**Syntax**

```
HrSameBlood(cGeneration,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSameBlood(cGeneration,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSameBlood(cGeneration,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Charater

## 16.4.31 HrSDam( ), JrSDam( ), SrSDam( )

The functions return the name of the horse's sire's dam.

**Syntax**

```
HrSDam( )
JrSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=22)

**See Also**

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

## 16.4.32 HrSDDam( ), JrSDDam( ), SrSDDam( )

The functions return the name of the horse's dam's dam.

**Syntax**

```
HrSDDam( )
JrSDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSDDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=22)

**See also**

© Welkin Systems Limited

HrSire( ), JrSire( ), SrSire( ) | HrDam( ), JrDam( ), SrDam( ) | HrSSire( ), JrSSire( ), SrSSire( ) | HrDSire( ), JrDSire( ), SrDSire( ) | HrSDam( ), JrSDam( ), SrSDam( ) | HrDDam( ), JrDDam( ), SrDDam( ) | HrSSSire( ), JrSSSire( ), SrSSSire( ) | HrDSSire( ), HrDSSire( ), SrDSSire( ) | HrSDSire( ), JrSDSire( ), SrSDSire( ) | HrDDSire( ), JrDDSire ( ), SrDDSire( ) | HrSSDam( ), JrSSDam( ), SrSSDam( ) | HrDSDam( ), JrDSDam( ), SrDSDam( ) | HrDDDam( ), JrDDDam( ), SrDDDam( )

## 16.4.33 HrSDSire( ), JrSDSire( ), SrSDSire( )

The functions return the name of the horse's sire's dam's sire.

### Syntax

```
HrSDSire( )
JrSDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSDSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=22)

### See Also

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

## 16.4.34 HrSex( ), JrSex( ), SrSex( )

Sex information of the horse. The function will return one of the following values:

- Filly/Mare
- Gelding
- Cold/Horse

### Syntax

```
HrSex( )
JrSex(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSec(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=11 in English mode, Width=6 in Chinese mode)

## 16.4.35 HrSickHistory( ), JrSickHistory( ), SrSickHistory( )

The functions return the veterinary records of the horse.

### Syntax 1

```
HrSickHistory(cKeywordList,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSickHistory(cKeywordList,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSickHistory(cKeywordList,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

© Welkin Systems Limited

### Syntax 2

```
HrSickHistory(nDays,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSickHistory(nDays,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSickHistory(nDays,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Syntax 3

```
HrSickHistory(cDayRange,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSickHistory(cDayRange,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSickHistory(cDayRange,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Syntax 4

```
HrSickHistory(cDayRangeKeywords,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSickHistory(cDayRangeKeywords,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSickHistory(cDayRangeKeywords,cFlags,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Arguments

cKeywordList

Returns veterinary records which contain keywords (case insensitive) in cKeywordList argument.

cFlags

| Veterinary Content | Show | Hide | Default |
|---|---|---|---|
| Incident date | D | -D | Show |
| Passed date | P | -P | Hide |
| Incident description | S | -S | Show |

nDays

Numeric argument. Returns veterinary records within pass nDays. A negative number for future days. 0 for the start day.

cDayRange

Character argument. Returns veterinary records with the day range specified. '1-90' for 90 days before start. '-2-90' for the period between 2 days after start and 90 days before start. '-30- -2' for the period between 30 days after start and 2 days after start.

cDayRangeKeywords

Character argument. Shows veterinary records within a period of time with a list of keywords found inside. Period will be specified in the format of cDayRange. Keywords will be put to the right of this day range text with a ":" sign between the day range text and keyword list.

### Returns

Multiline Text

### Remarks

In starter views, if you are going to pass a negative argument into these functions,

© Welkin Systems Limited

you are recommended to mark this function as "No Caching". It is because the cache will not be updated automatically unless you click on the [Recalc] button.

**Examples**

Returns all veterinary records of the horse:
```
HrSickHistory( )
```

Returns all veterinary records which contain the keywords "track work" and "jump":
```
HrSickHistory('track work,jump')
```

Returns all veterinary records with passed date, but hide the incident date.
```
HrSickHistory( ,'P-D')
```

Veterinary records with last 90 days
```
HrSickHistory(90)
```

Veterinary records after 2 days of the start
```
HrSickHistory(-2)
```

Veterinary records during the period 2 days after the start and 90 days before the start. (The day range must be in a pair of quotes)
```
HrSickHistory('-2-90')
```

Veterinary records during 90 days before the start with words either "blood" and "eye" found in text.
```
HrSickHistory('90:blood,eye')
```

**See Also**

HrBleedDate( ), HrBleedDateStr( )

## 16.4.36 HrSire( ), JrSire( ), SrSire( )

The functions return the name of the horse's sire.

**Syntax**

```
HrSire( )
JrSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=22)

**See Also**

HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

© Welkin Systems Limited

## 16.4.37 HrSSDam( ), JrSSDam( ), SrSSDam( )

The functions return the name of the horse's sire's sire's dam.

### Syntax

```
HrSSDam( )
JrSSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSSDam(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=22)

### See Also

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSire( ), JrSSire ( ), SrSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrDSDam( ), HrSDDam( ), HrDDDam( )

## 16.4.38 HrSSire( ), JrSSire( ), SrSSire( )

The functions return the name of the horse's sire's sire.

### Syntax

```
HrSSire( )
JrSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=22)

### See Also

HrSire( ), HrDam( ), HrDSire( ), HrSDam( ), HrDDam( ), HrSSSire( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

## 16.4.39 HrSSSire( ), JrSSSire( ), SrSSSire( )

The functions return the name of the horse's sire's sire's sire.

### Syntax

```
HrSSSire( )
JrSSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSSSire(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=22)

© Welkin Systems Limited

**See Also**

HrSire( ), HrDam( ), HrSSire( ), HrDSire( ), HrSDam( ), HrDDam( ), HrDSSire( ), HrDSSire( ), SrDSSire( ), HrSDSire( ), HrDDSire( ), HrSSDam( ), HrDSDam( ), HrSDDam ( ), HrDDDam( )

## 16.4.40 HrStable( ), JrStable( ), SrStable( )

The functions return the name of the stable of the horse. In Chinese language mode, Chinese horse name will be returned. In English language mode, English horse name will be returned.

**Syntax**

```
HrStable(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStable(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStable(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=8 in Chinese Mode, Width=30 in English Mode)

## 16.4.41 HrStableAbbr( ), JrStableAbbr( ), SrStableAbbr( )

Name abbreviation of stable. Please note:

1. Return values of the function is unique only for active stables and not for all stables
2. Return values of the function for some overseas stable may be blanks

**Syntax**

```
HrStableAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStableAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStableAbbr( )
```

**Returns**

Character (in English mode width=3, in Chinese mode width=2)

## 16.4.42 HrStableID( ), JrStableID( ), SrStableID( )

The functions return a unique character string identifying the stable.

**Syntax**

```
HrStableID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStableID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStableID( )
```

**Returns**

Character (Width=2)

© Welkin Systems Limited

### 16.4.43 HrStableRetired( ), JrStableRetired( ), SrStableRetired( )

Return whether the trainer is currently (not at his races) retired.

#### Syntax

```
HrStableRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrStableRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrStableRetired(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Logical

#### See Also

HrRetired( ), JrRetired( ), SrRetired( ); HrJockey( ), JrJockey( ), SrJockey( ); HrJockeyRetired( ), JrJockeyRetired( ), SrJockeyRetired( )

## 16.5 Track Work Information

This group of functions returns information about track work.

#### >>> Subtopics of this section <<<

- HrABTCount( ), JrABTCount( ), SrABTCount( )
- HrActivity( ), JrActivity( ), SrActivity( )
- HrTwResumeDate( ), JrTwResumeDate( ), SrTwResumeDate( )
- HrBestGallopSect( ), JrBestGallopSect( ), SrBestGallopSect( )
- HrBTrial( ), JrBTrial( ), SrBTrail( )
- HrGallop1200Best( ), JrGallop1200Best( ), SrGallop1200Best( )
- HrGallop1200Stat( ), JrGallop1200Stat( ), SrGallop1200Stat( )
- HrGallop400Best( ), JrGallop400Best( ), SrGallop400Best( )
- HrGallop400Stat( ), JrGallop400Stat( ), SrGallop400Stat( )
- HrGallop800Best( ), JrGallop800Best( ), SrGallop800Best( )
- HrGallop800Stat( ), JrGallop800Stat( ), SrGallop800Stat( )
- HrGBTCount( ), JrGBTCount( ), SrGBTCount( )
- HrRGBTCount( ), JrRGBTCount( ), SrRGBTCount( )
- HrTwCount( ), JrTwCount( ), SrTwCount( )

### 16.5.1 HrABTCount( ), JrABTCount( ), SrABTCount( )

The functions return the number of all weather track barrier trials during a number of days (specified by the first argument) before the underlying race.

#### Syntax

```
HrABTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
JrABTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
SrABTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
```

Note: nDays is default to 14.

© Welkin Systems Limited

### Returns

Numeric

### Examples

```
HrABTCount(7)
```

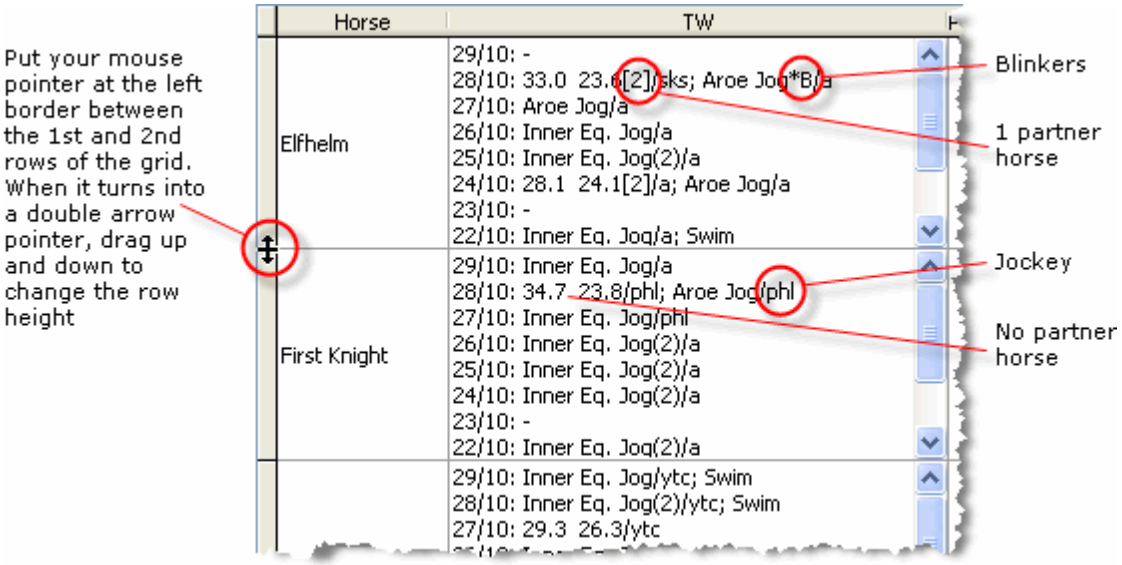
Returns the number of all weather track barrier trials during the last 7 days before the race. HrTwCount(7,'A') will return the same result except that HrTwCount(7,'A') returns characters, which may not be used in calculations.

### See also

HrTwCount( ), JrTwCount( ), SrTwCount( ) | HrABTCount( ), JrABTCount( ), SrABTCount( ) | HrGBTCount( ), JrGBTCount( ), SrGBTCount( ) | Occurs( )

## 16.5.2 HrActivity( ), JrActivity( ), SrActivity( )

Activities, including training and starts, of a horse in past N days. User comments for the activities will be displayed as well.


Put your mouse pointer at the left border between the 1st and 2nd rows of the grid. When it turns into a double arrow pointer, drag up and down to change the row height
Horse
TW
Elfhelm
29/10: -
28/10: 33.0 23.6[2]/sks; Aroe Jog*B/a
27/10: Aroe Jog/a
26/10: Inner Eq. Jog/a
25/10: Inner Eq. Jog(2)/a
24/10: 28.1 24.1[2]/a; Aroe Jog/a
23/10: -
22/10: Inner Eq. Jog/a; Swim
First Knight
29/10: Inner Eq. Jog/a
28/10: 34.7 23.8/phl; Aroe Jog/phl
27/10: Inner Eq. Jog/phl
26/10: Inner Eq. Jog(2)/a
25/10: Inner Eq. Jog(2)/a
24/10: Inner Eq. Jog(2)/a
23/10: -
22/10: Inner Eq. Jog(2)/a
29/10: Inner Eq. Jog/ytc; Swim
28/10: Inner Eq. Jog(2)/ytc; Swim
27/10: 29.3 26.3/ytc
Blinkers
1 partner horse
Jockey
No partner horse

### Syntax 1

```
HrActivity(nDays,cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVal,cInc
JrActivity(nDays,cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVal,cInc
SrActivity(nDays,cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVal,cInc
```

### Syntax 2

```
HrActivity('DayX-DayY',cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVa
JrActivity('DayX-DayY',cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVa
SrActivity('DayX-DayY',cTypeList,cChrForNA,cFlag,cSeparator,nBackRaces,'fCondition',cDefaultVa
```

### Arguments

© Welkin Systems Limited

| Argument | Description |
|---|---|
| nDays<br>(Syntax 1) | Number of days, e.g., 60.<br>If nDays is missing (or equals 0), the activities since last start will be returned. |
| 'DayX-DayY'<br>(Syntax 2) | Specifies a period from day X to day Y, counting back from the underlying race. For example '8-14' mean from day 8 to day 14 back from the underlying race. |
| cTypeList | A character string with list of activity types inside separated by ','.<br>Possible types:<br><br>RA - Race<br>BT - Barrier Trial<br>SW - Swim<br>JI - Inner AW/Eq Jog<br>JO - Outer AW/Eq Jog<br>JA - Aroe Jog<br>JT - Turf Jog<br><br>GL - Any type of galloping<br>GG - Turf track galloping<br>AG - All weather track galloping<br>RG - River side galloping |
| cChrForNA | Character which stands for no activity. Default to '-'. Pass .Null. into this argument to skip the lines without activities. |
| cFlag | Characters for specifying date format and suppressing user comments<br>N shows the day number from underlying start<br>D shows full date<br>S (or missing) dd/mm, e.g., 3rd of November will be shown as 03/11<br>U suppress user comments for the activity |
| cSeparator | Activity separator which is default to carriage return (Chr(13)). If you want to display all activities separated by a semicolon in the same line, you may pass '; ' into this argument. |
| nBackRaces | nBackRaces specifies activities before which start will be shown. For example, if nBackRaces=1 and nDays=30, the activities within 30 days before the last start (not current start) will be shown. nBackBaces is default to 0, i.e., the current start. |
| 'fCondition' | See 'MCL Condition' under 'Arguments of MCL functions' section |
| cDefaultVal | See 'The DefaultVal Argument' under 'Arguments of MCL functions' section |
| cIncluding | See 'The cIncluding Argument' under 'Arguments of MCL functions' section |

**Returns**

Character (variable length)

In views, normally, Multiline-Text type will be used for the column.

**Remark**

- Under normal circumstances, you may need to push the [Recalc] button to refresh the data of the column (see Cache for View Data for details).
- By default, user comments are displayed at the right hand side of the activity

© Welkin Systems Limited

descriptions. You may pass the flag "U" into the 4th argument (cFlag) to suppress the user comments.

**Examples**

Track work activities since last start:

```
HrActivity( )
```

Show jogging and swims since last start:

```
HrActivity( ,'JI,JO,JA,JT,SW')
```

Show galloping and barrier trials only in last 30 days. Use ' | ' (space, vertical bar, space) as separator between trials instead of putting the trials in separate lines.

```
HrActivity(30,'GL,BT',,,' | ')
```

Number of barrier trials during the week before last week:

```
HrActivity('8-14','BT')
```

**See Also**

HrTwCount( ), JrTwCount( ), SrTwCount( ) | HrABTCount( ), JrABTCount( ), SrABTCount( ) | HrGBTCount( ), JrGBTCount( ), SrGBTCount( ) | Occurs( )

### 16.5.3 HrTwResumeDate( ), JrTwResumeDate( ), SrTwResumeDate( )

The functions return the date on which the horse resumes its trackwork activities after the prior start. You may optionally pass one or more activity IDs into the first parameter to specify the type(s) of activity you are looking for. The function will then return the date on which this type of activity resumes.

**Syntax**

```
HrTwResumeDate(cActivityIDs,nBackRaces,'fCondition',nDefaultVal)
JrTwResumeDate(cActivityIDs,nBackRaces,'fCondition',nDefaultVal)
SrTwResumeDate(cActivityIDs,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Date

**Examples**

```
HrTwResumeDate( )
```

Returns the date on which track work activities (any type) resumes after the prior starts.

```
HrTwResumeDate('JLBO')
```

Returns the date on which jog (J), Gallop (L), Barrier Trial (B) or Jump-out (O) resumes since prior start.

© Welkin Systems Limited

### 16.5.4 HrBestGallopSect( ), JrBestGallopSect( ), SrBestGallopSect( )

The functions return the best galloping sectional (400m) time, in seconds, during a number of days before the race in which the horse runs.

#### Syntax

```
HrBestGallopSect(nDays,nBackRaces,'fCondition',nDefaultVal)
JrBestGallopSect(nDays,nBackRaces,'fCondition',nDefaultVal)
SrBestGallopSect(nDays,nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

#### Examples

```
HrBestGallopSect(7)
```

Returns the best sectional time, in second, during the last 7 days.

### 16.5.5 HrBTrial( ), JrBTrial( ), SrBTrail( )

The functions return the date and group of the barrier trial before the race. The draw number and finishing position are appended at the end of the returned string. There may be more than one barrier trials before a race. You can pass a number into the first argument to get the date of it. When you double-click on the column, the system will open the Barrier Trial form.

#### Syntax 1

```
HrBTrial(nBackBarrierTrials,nBackRaces,'fCondition',cDefaultVal)
JrBTrial(nBackBarrierTrials,nBackRaces,'fCondition',cDefaultVal)
SrBTrial(nBackBarrierTrials,nBackRaces,'fCondition',cDefaultVal)
```

#### Syntax 2

```
HrBTrial('Template',nBackRaces,'fCondition',cDefaultVal)
JrBTrial('Template',nBackRaces,'fCondition',cDefaultVal)
SrBTrial('Template',nBackRaces,'fCondition',cDefaultVal)
```

#### Argument

nBackBarrierTrials

Pass a number to the first argument to get the date and group of the *nBackBarrierTrials* (0 or 1 for first, 2 for second, etc) barrier trials before the underlying race.

Template

Template of the return string. You can include the following special symbols in the template. Corresponding item in the following table will be return in the position of the system.

| Special Symbol | Meaning / Corresponding Item |
|---|---|
| * | Return all barrier trials between last and current races |

© Welkin Systems Limited

| | |
|---|---|
| Numbers 0 to 9 | 1. No * in template: Number (n) in template stands for the the barrier trial n times before current race.<br>2. A * found in template: Number (n) in template stands for the all barrier trials between the n times and n+1 times starts back from the current start. If there is no barrier trial between these 2 starts, an empty string will be return. |
| U | User comments for the trial |
| P | Running positions (including finishing position) |
| F | Finishing position |
| D | Date |
| A | Draw position |
| N | Group |
| G | Gears |
| L | Distance |
| C | Course |
| T | Time to complete the run |
| J | Jockey |
| S | Sectional times |
| M | Margin behind the winner |

See Arguments of MCL functions for details about other arguments.

**Returns**

Character (Multiline Text)

**Examples**

The following functions return the date and group of the first barrier trial before the race.

```
HrBTrial()
HrBTrial(1)
```

Sample of return: "17/02/2009(5)" for the 5th group in 17/02/2009

The following function returns the date and group of the second barrier trial before the race.

```
HrBTrial(2)
```

The following functions return the date, jockey and finishing position of the latest trial.

```
HrBTrial('d j (f)')
HrBTrial('1d j (f)')              && '1' for last trial
```

Sample of return:
"05/03/2010 B Doyle (3)"

The following functions return the dates, jockeys and finishing positions of the trials between the current and last starts of the horses.

```
HrBTrial('*d j (f)')
HrBTrial('*0d j (f)')             && '0' for current start
```

© Welkin Systems Limited

Sample of return:
30/03/2010 M L Yeung (2)
16/03/2010 M L Yeung (8)

The following function returns the dates, jockeys and finishing positions of the trials between the last and second last starts of the horses.

```
HrBTrial('*1d j (f)')             && '0' for last start
```

Sample of return:
02/12/2008 H W Lai (6)
18/11/2008 G Mosse (3)

**See Also**

HrActivity( ), JrActivity( ), SrActivity( ), HrBTVideo( ), JrBTVideo( ), SrBTVideo( ), HrGBTCount( ), JrGBTCount( ), SrGBTCount( ), HrRGBTCount( ), JrRGBTCount( ), SrRGBTCount( )

## 16.5.6 HrGallop1200Best( ), JrGallop1200Best( ), SrGallop1200Best( )

The functions return the best time, in seconds, in all 1200m galloping during a number of days before the race in which the horse runs.

**Syntax**

```
HrGallop1200Best(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGallop1200Best(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGallop1200Best(nDays,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

## 16.5.7 HrGallop1200Stat( ), JrGallop1200Stat( ), SrGallop1200Stat( )

The functions return a character string showing:

- Number of 1200m gallops
- Best time in all 1200m gallops

during a number of days before the race in which the horse runs.

Information will be returned in the following format: 1.21.6 (4)

where 1.21.6 is the 'Best time' and 4 is the number of gallops.

**Syntax**

```
HrGallop1200Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGallop1200Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGallop1200Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Character (Width=12)

© Welkin Systems Limited

### 16.5.8 HrGallop400Best( ), JrGallop400Best( ), SrGallop400Best( )

The functions return the best time in all 400m galloping during a number of days before the race in which the horse runs.

**Syntax**

```
HrGallop400Best(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGallop400Best(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGallop400Best(nDays,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

### 16.5.9 HrGallop400Stat( ), JrGallop400Stat( ), SrGallop400Stat( )

The functions returns a character string showing:

- Number of 400m gallops
- Best time in all 400m gallops

during a number of days before the race in which the horse runs.

Information will be returned in the following format: 33.6 (4)

where 33.6 is the 'Best time' and 4 is the number of gallops.

**Syntax**

```
HrGallop400Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGallop400Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGallop400Stat(nDays,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Character (Width=12)

### 16.5.10 HrGallop800Best( ), JrGallop800Best( ), SrGallop800Best( )

The functions return the best time in all 800m galloping during a number of days before the race in which the horse runs.

**Syntax**

```
HrGallop800Best(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGallop800Best(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGallop800Best(nDays,nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

© Welkin Systems Limited

## 16.5.11 HrGallop800Stat( ), JrGallop800Stat( ), SrGallop800Stat( )

The functions returns a character string showing:

- Number of 800m galloping
- Best time in all 800m gallopings

during a number of days before the race in which the horse runs.

Information will be returned in the following format: 53.6 (4)

where 53.6 is the 'Best time' and 4 is the number of gallops.

### Syntax

```
HrGallop800Stat(nDays,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrGallop800Stat(nDays,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrGallop800Stat(nDays,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=12)

## 16.5.12 HrGBTCount( ), JrGBTCount( ), SrGBTCount( )

The above functions return the number of grass barrier trials, excluding riverside grass barrier trials, during a number of days (specified by the first argument) before the underlying race.

### Syntax

```
HrGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
JrGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
SrGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
```

Note: nDays is default to 14.

### Returns

Numeric

### Examples

```
HrGBTCount(7)
```

Returns the number of turf barrier trials during the last 7 days before the race. HrTwCount(7,'G') will return the same result except that HrTwCount(7,'G') returns characters, which may not be used in calculations.

### See also

HrABTCount( ), JrABTCount( ), SrABTCount( ) | HrTwCount( ), JrTwCount( ), SrTwCount( )

© Welkin Systems Limited

## 16.5.13 HrRGBTCount( ), JrRGBTCount( ), SrRGBTCount( )

The above functions return the number of river side grass barrier trials, excluding riverside grass barrier trials, during a number of days (specified by the first argument) before the underlying race.

### Syntax

```
HrRGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
JrRGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
SrRGBTCount(nDays,nBackRaces,'fCondition',nDefaultVal)
```

Note: nDays is default to 14.

### Returns

Numeric

### Examples

```
HrRGBTCount(7)
```

Returns the number of river side turf barrier trials during the last 7 days before the race. HrTwCount(7,'R') will return the same result except that HrTwCount(7,'R') returns characters, which may not be used in calculations.

### See also

HrABTCount( ), HrTwCount( ), HrGBTCount( )

## 16.5.14 HrTwCount( ), JrTwCount( ), SrTwCount( )

The functions return the statistics on activities of a horse during a number of days (specified by the first argument) before the underlying race. You may specify which types of activities you want by passing a list of control symbols into the second argument of the function. The following figure shows how data are displayed in a column.

| Horse | Gp,Jg,Sw / Ib,Ab,Gb |
|---|---|
| Art Trader | 5,17,17 / , , |
| Winning Kevin | 1,12,13 / 1, , |
| California Bishop | 19,76, 1 / , 1, |
| Industrial Succes | 3, 9, / , , |
| Northerner | 11,71, / , 1, 1 |
| Fortune Smiles | 2,12, / , , |
| Fighting Mascot | 4,12, / , , |
| Dashing Champio | 3,10,14 / , , |
| Millennium Spirit | 1, 8, 4 / , , |
| Gallant Knight | 2, 6, 7 / , , |
| Stable Mail | 1, 9, / 1, , |
| King Of Fish | 6,68,90 / 2, , |

### Syntax 1

```
HrTwCount(nDays,cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
JrTwCount(nDays,cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
SrTwCount(nDays,cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
```

© Welkin Systems Limited

### Syntax 2

```
HrTwCount('DayX-DayY',cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
JrTwCount('DayX-DayY',cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
SrTwCount('DayX-DayY',cTemplate,cZeroMark,nBackRaces,'fCondition',nDefaultVal)
```

### Argument

`nDays (Syntax 1)`

Number of days, e.g., 60.

If nDays is missing (or equals 0), the activities since last start will be returned.

'DayX-DayY' (Syntax 2)

Specifies a period from day X to day Y, counting back from the underlying race. For example '8-14' mean from day 8 to day 14 back from the underlying race.

cTemplate

A template containing the following control symbols which control the return value of the function:

| Symbols | Returns | Remark |
|---|---|---|
| G g | Number of turf barrier trials | Capital letters for formal barrier trials while small letters for Jump-out |
| R r | Number of river-side turf barrier trials | |
| A a | Number of all weather track barrier trials | |
| L1 | Gallop (no partner) | |
| L2 | Gallop (1 partner) | |
| L3 | Gallop (2 or more partners) | |
| L | Gallop (with or without partners) | |
| J | Jog | |
| S | Swim | |
| C | Start | |
| P | Spelling | |
| N | No activity (including spelling days) | |
| I | No activity (excluding spelling days) | |

- In the template, you may separate different types of Activities by commas or other punctuation marks. More than one type of activity may be grouped together by putting more than one symbol together.
- cReturnList is default to 'L,J,S /gra,A,G'

cZeroMark

A character string which will replace the zero value in the returned value list. If you don't pass a character string into this parameter, zero values will be replaced with blank spaces.

### Returns

Character (width is template dependent; 31 for the default template)

© Welkin Systems Limited

**Remark**

Under normal circumstances, you may need to push the [Recalc] button to refresh the data of the column (see Cache for View Data for details).

**Examples**

```
HrTwCount( ,'GA,gra,L,J,S','- ')
```

'GA' stands for total number of formal barrier trials. 'gra' stands for the total number of jump-outs. L, J, S stand for gallop, jog and swim respectively. If we apply the function to race number 9 of 17 Sept 2005, the following list will be returned for the horse Chinsola:

" 1,- , 3,32,- "  (1 format barrier trial, 0 jump-out, 3 gallops, 32 jogs, 0 swim)

```
HrTwCount('425-453','GA,gra,L,J,S','- ')
```

Similar to the example above. The only difference is the statistical period is specified as 453 days ago to 424 days ago (30 days in total) from the underlying race.

```
HrTwCount( ,'GA,gra,L,J,S','0')
```

We will get the same values as shown in the above example except that zero will be shown as '0':

" 1, 0, 3,32, 0"

```
Trim(HrTwCount(28,'gra,A,G'),' ',',')
```

The functions in this example will return the counts of all jump-outs, formal AW barrier trials and turf barrier trials in last 28 days. The Trim( ) function removes all trailing spaces (' ') and commas (',').

```
HrTwCount(14,'G,gr')
```

The function will return the number of formal and turf jump-outs in last 14 days.

```
HrTwCount(30,'L1,L2L3,L')
HrTwCount(30,'L1,L2L3,L1L2L3')
```

The 2 functions above will return the same result, which is the numbers of single-horse gallops (no partner), partnered gallops, and all gallops in last 30 days. The sum of the first 2 figures (L1 and L2L3) will be equal to the 3rd figure (L or L1L2L3).

**See also**

HrActivity( ), JrActivity( ), SrActivity( ) | HrABTCount( ), JrABTCount( ), SrABTCount ( ) | HrGBTCount( ), JrGBTCount( ), SrGBTCount( )

## 16.6 Race and Meeting Information

This group of functions returns particulars about a race.

© Welkin Systems Limited

**>>> Subtopics of this section <<<**

- ComingDate( )
- CurRunHorses( )
- DateOfRace( )
- HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( )
- HrAccSectionalTimew( ), JrAccSectionalTimew( ), SrAccSectionalTimew( )
- HrAveRating( ), JrAveRating( ), SrAveRating( )
- HrClassGroup( ), JrClassGroup( ), SrClassGroup( )
- HrCourse( ), JrCourse( ), SrCourse( )
- HrCourseDist( ), JrCourseDist( ), SrCourseDist( )
- HrCourseID( ), JrCourseID( ), SrCourseID( )
- HrCupName( ), JrCupName( ), SrCupName( )
- HrDate( ), JrDate( ), SrDate( )
- HrDayNight( ), JrDayNight( ), SrDayNight( )
- HrDayNote( ), JrDayNote( ), SrDayNote( )
- HrDaysAgo( ), JrDaysAgo( ), SrDaysAgo( )
- HrDistance( ), JrDistance( ), SrDistance( )
- HrDistTrack( ), JrDistTrack( ), SrDistTrack( )
- HrDvd( ), JrDvd( ), SrDvd( )
- HrDvdPlace( ), JrDvdPlace( ), SrDvdPlace( )
- HrDvdTrio( ), JrDvdTrio( ), SrDvdTrio( )
- HrDvdWin( ), JrDvdWin( ), SrDvdWin( )
- HrGoing( ), JrGoing( ), SrGoing( )
- HrGoingAbbr( ), JrGoingAbbr( ), SrGroingAbbr( )
- HrGoingID( ), JrGoingID( ), SrGoingID( )
- HrIsTurf( ), JrIsTurf( ), SrIsTurf( )
- HrLastSectionTime( ), JrLastSectionTime( ), SrLastSectionTime( )
- HrLastSectionTimew( ), JrLastSectionTimew( ), SrLastSectionTimew( )
- HrMeetingNo( ), JrMeetingNo( ), SrMeetingNo( )
- HrPenetrometer( ), JrPenetrometer( ), SrPenetrometer( )
- HrRaceClass( ), JrRaceClass( ), SrRaceClass( )
- HrRaceID( ), JrRaceID( ), SrRaceID( )
- HrRaceNoInMeeting( ), JrRaceNoInMeeting( ), SrRaceNoInMeeting( )
- HrRaceNoInSeason( ), JrRaceNoInSeason( ), SrRaceNoInSeason( )
- HrRaceNote( ), JrRaceNote( ), SrRaceNote( )
- HrRacePrize( ), JrRacePrize( ), SrRacePrize( )
- HrRaceRating( ), JrRaceRating( ), SrRaceRating( )
- HrRaceRemark( ), JrRaceRemark( ), SrRaceRemark( )
- HrRacesInMeeting( ), JrRacesInMeeting( ), SrRacesInMeeting( )
- HrRaceTime( ), JrRaceTime( ), SrRaceTime( )
- HrRunners( ), JrRunners( ), SrRunners( )
- HrSeason( ), JrSeason( ), SrSeason( )
- HrSeason2( ), JrSeason2( ), SrSeason2( )
- HrSeasonRace( ), JrSeasonRace( ), SrSeasonRace( )
- HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( )
- HrSectionalTimew( ), JrSectionalTimew( ), SrSectionalTimew( )
- HrSectionTime( ), JrSectionTime( ), SrSectionTime( )
- HrSectionTimew( ), JrSectionTimew( ), SrSectionTimew( )
- HrPowerFigure( ), JrPowerFigure( ), SrPowerFigure( )
- HrPowerComments( ), JrPowerComments( ), SrPowerComments( )
- HrTrack( ), JrTrack( ), SrTrack( )
- HrTrackID( ), JrTrackID( ), SrTrackID( )
- HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( )
- HrTrackShift( ), JrTrackShift( ), SrTrackShift( )
- HrWinner( ), JrWinner( ), SrWinner( )
- HrWinnerID( ), JrWinnerID( ), SrWinnerID( )

© Welkin Systems Limited

- HrWinSeconds( ), JrWinSeconds( ), SrWinSeconds( )
- HrWinTime( ), JrWinTime( ), SrWinTime( )
- HrWinTimew( ), JrWinTimew( ), SrWinTimew( )
- IrregularRace( )

### 16.6.1 ComingDate( )

The function returns the date of the coming race meeting, which has not yet started. If there is no coming meeting in the database date of the latest started meeting will be returned.

#### Syntax

```
ComingDate( )
```

#### Returns

Date

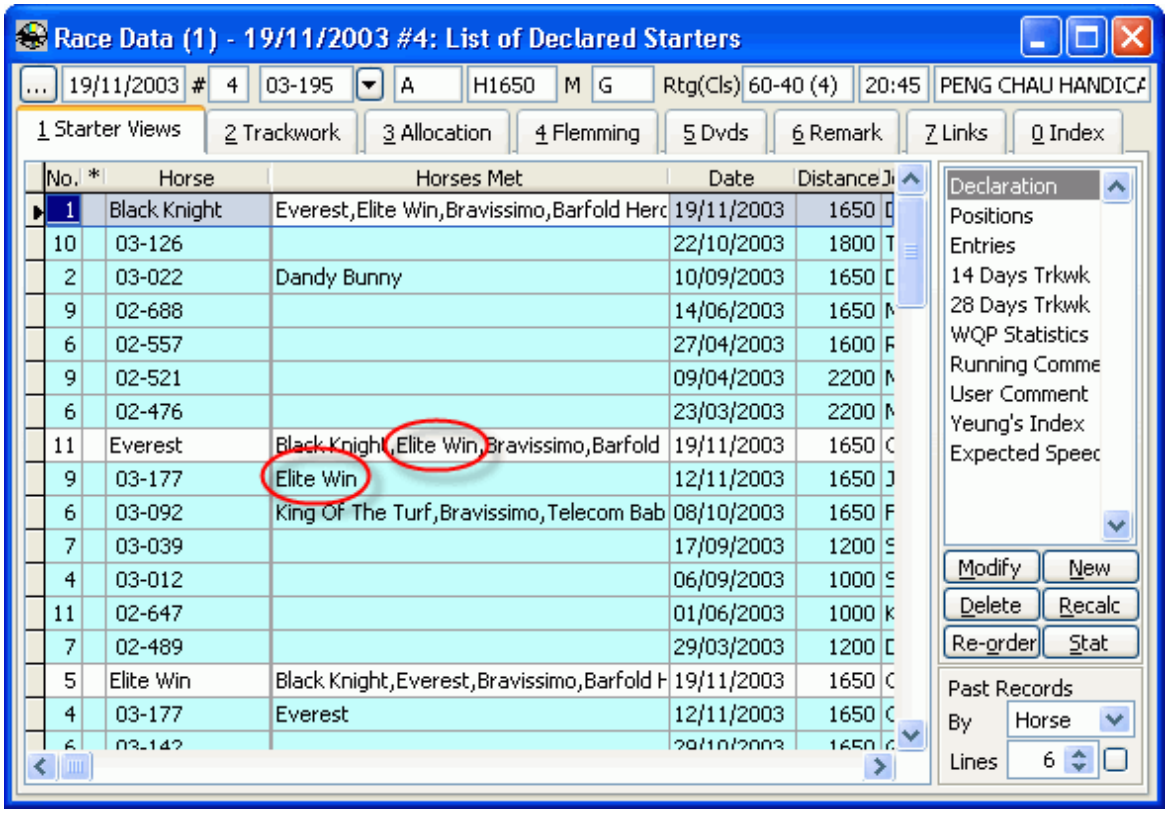
#### See also

HrDate( )

### 16.6.2 CurRunHorses( )

The function returns a comma separated list of starter name of the current run (viewing race). When you show past records of a horse in a view, the function will filter from the past races the horses which participate in the current race. The example view in the figure below shows data of race 195 of season 03-04. The CurRunHorse( ) function in the 4th column shows that Everest had met Elite Win 03-177.

© Welkin Systems Limited




Please note the following limitation of the functions:

1. CurRunHorse( ) function will return .NULL. in statistics and model evaluation since "viewing race" will be undefined when you evaluate a model (clicking the "Stat" button in the Race Meeting form).
2. Since viewing race changes whenever you jump to a new race, cached old values returned by CurRunHorse( ) in history lines will be out-of-date all the time. You are recommended to set the column type to "No Caching", otherwise you will need to push the "Recalc" button to update the values whenever a new race is displayed.

### Syntax

```
CurRunHorses(cExpr,cSeperator)
```

### Argument

cExpr

Optional. Can be an MCL function or any other expression, which is to replace the horse name and be shown on the races which the horse(s) meets the underlying horse.

cSeperator

Optional. Default to ", "

### Returns

Character (Number of character returned will vary. You are recommended to set column data type to "Multiline Text")

© Welkin Systems Limited

### 16.6.3 DateOfRace( )

Returns the date of a race.

#### Syntax

```
DateOfRace(cSeasonRace)
```

#### Argument

cSeasonRace

Optional argument with format "SSSS-N", where SSSS stands for season and N stands for race index. If the argument is omitted, the underlying race is assumed.

#### Returns

Date

#### Example

```
DateOfRace( )
```

Returns the date of the underlying race.

```
DateOfRace('2010-118')
```

Returns the date on which race 118 in season 2010-2011 takes place.

### 16.6.4 HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( )

Accumulated sectional times. This function group is similar to the HrSectionalTime( ) function group. Both groups will return the sectional times of a race in a string. The difference is that every sectional time return from this group has already been added the total time of all preceding sections.

#### Syntax

```
HrAccSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrAccSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrAccSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (Width=38)

#### Sample Return Value

"0.24.1 0.47.8 1.11.2 "

#### See also

HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( ) | HrSectionTime( ), JrSectionTime( ), SrSectionTime( )

© Welkin Systems Limited

### 16.6.5 HrAccSectionalTimew( ), JrAccSectionalTimew( ), SrAccSectionalTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( )

**Returns**

Character (Width=44)

### 16.6.6 HrAveRating( ), JrAveRating( ), SrAveRating( )

The functions return the average rating of all the horses in the race in which the horse runs.

**Syntax**

```
HrAveRating(nBackRaces,'fCondition',nDefaultVal)
JrAveRating(nBackRaces,'fCondition',nDefaultVal)
SrAveRating(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

### 16.6.7 HrClassGroup( ), JrClassGroup( ), SrClassGroup( )

The functions return the class group of the underlying race.

**Syntax**

```
HrClassGroup(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrClassGroup(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrClassGroup(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=5)

**See also**

HrRaceClass( ), JrRaceClass( ), SrRaceClass( ) | HrClass( ), JrClass( ), SrClass( ) | HrClassID( ), JrClassID( ), SrClassID( )

### 16.6.8 HrCourse( ), JrCourse( ), SrCourse( )

The functions return the course of the race.

| Course | Returned value in English | Returned value in Chinese |
|---|---|---|

© Welkin Systems Limited

| | Mode | Mode |
|---|---|---|
| Happy Valley | HV | 谷 |
| Shatin | SH | 田 |

**Syntax**

```
HrCourse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrCourse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCourse(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=2 in English mode, Width=2 in Chinese mode)

## 16.6.9 HrCourseDist( ), JrCourseDist( ), SrCourseDist( )

The functions return a string showing the course and distance of the race.

**Syntax**

```
HrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=6)

**Examples of Returned Value**

| English Mode | Chinese Mode | For |
|---|---|---|
| "H1650 " | "谷1650" | Happy valley 1650m |
| "S1400 " | "田1400" | Shatin 1400m |

## 16.6.10 HrCourseID( ), JrCourseID( ), SrCourseID( )

The functions return the ID of the racecourse.

| Course | ID |
|---|---|
| Happy Valley | H |
| Shatin | S |

**Syntax**

```
HrCourseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrCourseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCourseID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

© Welkin Systems Limited

Character (Width=1)

### 16.6.11 HrCupName( ), JrCupName( ), SrCupName( )

The functions return the cup name of a race.

#### Syntax

```
HrCupName(nBackRaces,'fCondition',dDefaultVal)
JrCupName(nBackRaces,'fCondition',dDefaultVal)
SrCupName(nBackRaces,'fCondition',dDefaultVal)
```

#### Returns

Character (Width=18 in Chinese mode, Width=30 in English mode)

### 16.6.12 HrDate( ), JrDate( ), SrDate( )

The functions return the date of start of the horse.

#### Syntax

```
HrDate(nBackRaces,'fCondition',dDefaultVal)
JrDate(nBackRaces,'fCondition',dDefaultVal)
SrDate(nBackRaces,'fCondition',dDefaultVal)
```

#### Returns

Date

#### See Also

HrRaceTime( ), JrRaceTime( ), SrRaceTime( )

### 16.6.13 HrDayNight( ), JrDayNight( ), SrDayNight( )

The functions return a character showing which time period in a day the meeting takes place.

#### Syntax

```
HrDayNight(nBackRaces,'fCondition',nDefaultVal)
JrDayNight(nBackRaces,'fCondition',nDefaultVal)
SrDayNight(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Character (Width=1)

| Returns | Meaning |
|---|---|
| D | Day Meeting |

© Welkin Systems Limited

| | |
|---|---|
| N | Evening Meeting |
| E | Twilight Meeting |

### 16.6.14 HrDayNote( ), JrDayNote( ), SrDayNote( )

The functions return the your personal notes for the meeting. The notes are input on the second page of the Race Calendar form.

#### Syntax

```
HrDayNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrDayNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDayNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (width equals number of characters of the note)

#### See Also

HrRaceRemark( ), JrRaceRemark( ), SrRaceRemark( ) | HrRaceNote( ), JrRaceNote( ), SrRaceNote( )

### 16.6.15 HrDaysAgo( ), JrDaysAgo( ), SrDaysAgo( )

The functions return how many days ago, relative to the underlying race, the race takes place.

#### Syntax

```
HrDaysAgo(nBackRaces,'fCondition',nDefaultVal)
JrDaysAgo(nBackRaces,'fCondition',nDefaultVal)
SrDaysAgo(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

### 16.6.16 HrDistance( ), JrDistance( ), SrDistance( )

The functions return the distance of the race.

#### Syntax

```
HrDistance(nBackRaces,'fCondition',nDefaultVal)
JrDistance(nBackRaces,'fCondition',nDefaultVal)
SrDistance(nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

Numeric

© Welkin Systems Limited

**Example of Returned Value**

1650

## 16.6.17 HrDistTrack( ), JrDistTrack( ), SrDistTrack( )

The functions return a string showing the course and distance of the race. The last 2 characters of the return string are for course-track.

**Syntax**

```
HrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrCourseDist(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=6 in English mode; Width=8 in Chinese mode)

**Examples of Returned Value**

| English Mode | Chinese Mode | For |
|---|---|---|
| "1650HG" | "1650谷草" | Happy Turf valley 1650m |
| "1400SG" | "1400田草" | Shatin Turf 1400m |
| "1600SA" | "1600田全" | Shatin All-Weather 1600m |
| "1400MG" | "1400澳草" | Turf 1400m |
| "1200MS" | "1200澳沙" | Sand 1200m |

**See also**

HrDistance( ), JrDistance( ), SrDistance( ) | HrCourseDist( ), JrCourseDist( ), SrCourseDist( ) | HrTrack( ), JrTrack( ), SrTrack( ) | HrCourse( ), JrCourse( ), SrCourse( )

## 16.6.18 HrDvd( ), JrDvd( ), SrDvd( )

The function the amount of a dividend. If the dividend you specified does not exist (or the instance does not exist), the function will return .NULL.

**Syntax**

```
HrDvd(cDividendID,nInstance,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrDvd(cDividendID,nInstance,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDvd(cDividendID,nInstance,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Argument**

cDividendID

A symbol (case insensitive) which represents a dividend according to the following table.

© Welkin Systems Limited

| DividendID | Dividend Title |
|---|---|
| WIN | Win |
| PLA | Place |
| QIN | Quinella |
| QPL | Quinella Place |
| TCE | Tierce |
| TRI | Trio |
| D-Q | Double Quinella |
| D-T | Double Trio |
| T-T | Triple Trio |
| DBL | Double |
| DBLC | Double Consolation |
| TBL | Treble |
| TBLC | Treble Consolation |
| 6UP | Six Up |
| 6UPB | Six Up Win Bonus |
| 5WN | Five Win |
| F-F | First Four |

nInstance

Instance number of a dividend. Default to 1.

**Return**

Numeric (Decimal=1)

**Examples**

Under normal circumstance, there will be 3 place dividends. They can be retrieved by the following functions:

First place dividend: `HrDvd('PLA',1) or HrDvd('PLA')`

Second place dividend: `HrDvd('PLA',2)`

Third place dividend: `HrDvd('PLA',3)`

## 16.6.19 HrDvdPlace( ), JrDvdPlace( ), SrDvdPlace( )

The function returns place dividend. This function is for backward compatible only. Please use HrDvd( ) instead.

**Syntax**

```
HrDvdPlace(nPlaceNo,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrDvdPlace(nPlaceNo,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDvdPlace(nPlaceNo,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

The function returns the first place dividend if nPlaceNo=1, second place dividend if nPlaceNo=2, and third place dividend if nPlaceNo=3.

© Welkin Systems Limited

**Returns**

Numeric (decimal place=1)

**See also**

HrDvdWin( ), JrDvdWin( ), SrDvdWin( )

## 16.6.20 HrDvdTrio( ), JrDvdTrio( ), SrDvdTrio( )

The functions return the Trio dividend. This function is for backward compatible only. Please use HrDvd( ) instead.

**Syntax**

```
HrDvdTrio(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrDvdTrio(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDvdTrio(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (decimal place=1)

**See also**

HrDvdPlace( ), JrDvdPlace( ), SrDvdPlace( ) | HrDvdWin( ), JrDvdWin( ), SrDvdWin( )

## 16.6.21 HrDvdWin( ), JrDvdWin( ), SrDvdWin( )

The function returns win dividend. This function is for backward compatible only. Please use HrDvd( ) instead.

**Syntax**

```
HrDvdWin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrDvdWin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrDvdWin(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (decimal place=1)

**See Also**

HrDvdPlace( ), JrDvdPlace( ), SrDvdPlace( )

## 16.6.22 HrGoing( ), JrGoing( ), SrGoing( )

The functions return the going of the race in which the horse runs.

© Welkin Systems Limited

**Syntax**

```
HrGoing(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrGoing(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrGoing(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=20 in English mode, Width=4 in Chinese mode)

**See Also**

HrGoingID( ), HrIsTurf( )

### 16.6.23 HrGoingAbbr( ), JrGoingAbbr( ), SrGroingAbbr( )

The functions return an abbreviation for each going.

**Syntax**

```
HrGoingAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrGoingAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrGoingAbbr(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=2)

| Abbreviation | Going |
|---|---|
| F | Firm |
| GF | Good to Firm |
| G | Good |
| GY | Good to Yielding |
| Y | Yielding |
| SY | Soft to Yielding |
| S | Soft |
| H | Heavy |
| Sn | Sand |
| SR | Sand (Raining) |
| Q | Equitrack |
| QR | Equitrack (Raining) |
| AW | All Weather Track |
| AR | All Weather (Raining) |
| WF | Wet Fast |
| FA | Fast All Weather |
| GA | Good All Weather |
| SA | Slow All Weather |
| WS | Wet Slow All Weather |

© Welkin Systems Limited

**See also**

HrGoing( ), JrGoing( ), SrGoing( ) | HrGoingID( ), JrGoingID( ), SrGoingID( )

## 16.6.24 HrGoingID( ), JrGoingID( ), SrGoingID( )

The functions return the internal ID of race goings in which the horse runs.

**Syntax**

```
HrGoingID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrGoingID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrGoingID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

| ID | Going |
|---|---|
| 1 | Firm |
| 2 | Good to Firm |
| 3 | Good |
| 4 | Good to Yielding |
| 5 | Yielding |
| 6 | Soft to Yielding |
| 7 | Soft |
| 8 | Heavy |
| 9 | Sand |
| A | Sand (Raining) |
| B | Equitrack |
| C | Equitrack (Raining) |
| E | All Weather Track |
| F | All Weather (Raining) |
| G | Wet Fast |
| H | Fast All Weather |
| I | Good All Weather |
| J | Slow All Weather |
| K | Wet Slow All Weather |

**See also**

HrGoing( ), JrGoing( ), SrGoing( ) | HrGoingAbbr( ), JrGoingAbbr( ), SrGroingAbbr( )

## 16.6.25 HrIsTurf( ), JrIsTurf( ), SrIsTurf( )

If the race is on a turf track, the functions return .T. (true). Otherwise the functions return .F. (false).

© Welkin Systems Limited

#### Syntax

```
HrIsTurf(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrIsTurf(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrIsTurf(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Logical

#### See Also

HrGoing( ), HrGoingID( )

### 16.6.26 HrLastSectionTime( ), JrLastSectionTime( ), SrLastSectionTime( )

The functions return the time (in seconds) of the section, which is counted starting from the last section, specified by the 1st argument.

#### Syntax

```
HrLastSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrLastSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrLastSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric (Characters=4, Decimals=1)

#### Example of Returned Value

24.2

#### Examples

```
HrLastSectionTime( )
```

Or

```
HrLastSectionTime(1)
```

The above 2 functions return the same value, i.e., time (in seconds) for the last section. Please note that the "1" in the brackets is for 1st last section rather than the prior start.

```
HrLastSectionTime( ,1)
```

Or

```
HrLastSectionTime(1,1)
```

The above 2 functions return the same value, i.e., time (in seconds) for the last section of the race of the last start. Please note that the last "1" in the brackets refers to the prior start the horse.

```
HrLastSectionTime(2)
```

Returns time for the 2nd last section of the race.

© Welkin Systems Limited

**See also**

HrSeconds( ), JrSeconds( ), SrSeconds( ) | HrSectionTime( ), JrSectionTime( ), SrSectionTime( ) | HrLastSectPos( ), JrLastSectPos( ), SrLastSectPos( )

## 16.6.27 HrLastSectionTimew( ), JrLastSectionTimew( ), SrLastSectionTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrLastSectionTime( ), JrLastSectionTime( ), SrLastSectionTime( )

**Returns**

Numeric (Characters=5, Decimals=2)

## 16.6.28 HrMeetingNo( ), JrMeetingNo( ), SrMeetingNo( )

The functions return a start's race number in a meeting.

**Syntax**

```
HrMeetingNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrMeetingNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrMeetingNo(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (Width=4)

## 16.6.29 HrPenetrometer( ), JrPenetrometer( ), SrPenetrometer( )

The functions return the Penetrometer and Clegg Hammer Readings of the race course.

**Syntax**

```
HrPenetrometer(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPenetrometer(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPenetrometer(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=25)

**See also**

HrCourse( ), JrCourse( ), SrCourse( )

© Welkin Systems Limited

## 16.6.30 HrRaceClass( ), JrRaceClass( ), SrRaceClass( )

The functions return the class of the underlying race.

### Syntax

```
HrRaceClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceClass(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=3)

### See also

Introduction to RaceMate | HrClass( ), JrClass( ), SrClass( ) | HrClassID( ), JrClassID ( ), SrClassID( )

## 16.6.31 HrRaceID( ), JrRaceID( ), SrRaceID( )

The functions return a unique character string representing a particular race of a horse.

### Syntax

```
HrRaceID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=5)

### Example of Returned Value

| Returned Value | Description |
|---|---|
| "18023" | The 23 rd races in the season 1998-1999 |

## 16.6.32 HrRaceNoInMeeting( ), JrRaceNoInMeeting( ), SrRaceNoInMeeting( )

The functions return a start's race number in a meeting.

### Syntax

```
HrRaceNoInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceNoInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceNoInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Numeric (Width=2)

© Welkin Systems Limited

### 16.6.33 HrRaceNoInSeason( ), JrRaceNoInSeason( ), SrRaceNoInSeason( )

The functions return a start's race number in a season.

#### Syntax

```
HrRaceNoInSeason(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceNoInSeason(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceNoInSeason(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric (Width=3)

### 16.6.34 HrRaceNote( ), JrRaceNote( ), SrRaceNote( )

The functions return the your personal notes for the race. The notes are input on the Comment page (6th page) of the Race Data form.

#### Syntax

```
HrRaceNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceNote(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Character (width equals number of characters of the note)

#### See Also

HrRaceRemark( ), JrRaceRemark( ), SrRaceRemark( ) | HrDayNote( ), JrDayNote( ), SrDayNote( )

### 16.6.35 HrRacePrize( ), JrRacePrize( ), SrRacePrize( )

The functions return the prize of a race. Please note that prize information is only available after September of 2008.

#### Syntax

```
HrRacePrize(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRacePrize(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRacePrize(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

#### Returns

Numeric

#### See Also

HrStakes( )

© Welkin Systems Limited

### 16.6.36 HrRaceRating( ), JrRaceRating( ), SrRaceRating( )

The functions return the rating range of a race.

**Syntax**

```
HrRaceRating(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceRating(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceRating(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=7)

**See also**

HrRating( ), JrRating( ), SrRating( )

### 16.6.37 HrRaceRemark( ), JrRaceRemark( ), SrRaceRemark( )

The functions return the Jockey Club remarks for a race.

**Syntax**

```
HrRaceRemark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRaceRemark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRaceRemark(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width equals number of characters of the remark)

### 16.6.38 HrRacesInMeeting( ), JrRacesInMeeting( ), SrRacesInMeeting( )

The functions return number of races of a meeting.

**Syntax**

```
HrRacesInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrRacesInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrRacesInMeeting(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (Width=2)

### 16.6.39 HrRaceTime( ), JrRaceTime( ), SrRaceTime( )

The functions return the commencing time (in 24-hour format) of a race

**Syntax**

```
HrRaceTime(nBackRaces,'fCondition',dDefaultVal)
```

© Welkin Systems Limited

```
JrRaceTime(nBackRaces,'fCondition',dDefaultVal)
SrRaceTime(nBackRaces,'fCondition',dDefaultVal)
```

**Returns**

Character (Width=5)

**Example of Returned Value**

"19:00"

**See also**

HrData( ), JrData( ), SrData( )

## 16.6.40 HrRunners( ), JrRunners( ), SrRunners( )

The functions return the number of horses in the race in which the horse runs.

**Syntax**

```
HrRunners(nBackRaces,'fCondition',nDefaultVal)
JrRunners(nBackRaces,'fCondition',nDefaultVal)
SrRunners(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**Example of Returned Value**

| Returned Value | Description |
|---|---|
| 13 | 13 runners in the race |

## 16.6.41 HrSeason( ), JrSeason( ), SrSeason( )

The above function returns race season of the underlying race. A returned value 2000 shows that the race is in season 2000-2001, and so on.

**Syntax**

```
HrSeason(nBackRaces,'fCondition',nDefaultVal)
JrSeason(nBackRaces,'fCondition',nDefaultVal)
SrSeason(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Numeric

**Example of Returned Value**

| Returned Value | Description |
|---|---|

© Welkin Systems Limited

| | |
|---|---|
| 2000 | Season 2000-2001 |
| 2001 | Season 2001-2002 |

**See also**

HrSeason2( ), JrSeason2( ), SrSeason2( ) | HrSeasonRace( ), JrSeasonRace( ), SrSeasonRace( )

## 16.6.42 HrSeason2( ), JrSeason2( ), SrSeason2( )

The above function returns the race season of the underlying race by joining the starting and ending years with a hyphen. Years are 2-digit numbers.

**Syntax**

```
HrSeason2(nBackRaces,'fCondition',nDefaultVal)
JrSeason2(nBackRaces,'fCondition',nDefaultVal)
SrSeason2(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Character (Width=5)

**Example of Returned Value**

| Returned Value | Description |
|---|---|
| 00-01 | Season 2000-2001 |
| 01-02 | Season 2001-2002 |

**See also**

HrSeason( ), JrSeason( ), SrSeason( ) | HrSeasonRace( ), JrSeasonRace( ), SrSeasonRace( )

## 16.6.43 HrSeasonRace( ), JrSeasonRace( ), SrSeasonRace( )

The functions return a string showing the racing season and the race number in the season. 2000-214 stands for the 214th race in season 2000-2001

**Syntax**

```
HrSeasonRace(nBackRaces,'fCondition',nDefaultVal)
JrSeasonRace(nBackRaces,'fCondition',nDefaultVal)
SrSeasonRace(nBackRaces,'fCondition',nDefaultVal)
```

**Returns**

Character (Width=9)

© Welkin Systems Limited

## 16.6.44 HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( )

The above functions return the sectional time of a race.

### Syntax

```
HrSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectionalTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=34)

### Example of Returned Value

"24.1 23.7 23.4 "

### See also

HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( ) | HrSectionTime ( ), JrSectionTime( ), SrSectionTime( )

## 16.6.45 HrSectionalTimew( ), JrSectionalTimew( ), SrSectionalTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( )

### Returns

Character (Width=40)

## 16.6.46 HrSectionTime( ), JrSectionTime( ), SrSectionTime( )

The functions return the time of an individual section, specified by the first argument (nSection) of the function, of a race.

### Syntax

```
HrSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSectionTime(nSection,nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Numeric (Width=4, Decimal=1)

### Examples

```
HrSectionTime( )
```

Or

© Welkin Systems Limited

```
HrSectionTime(1)
```

The above 2 functions return the same value, i.e., time (in seconds) for the first section. Please note that the "1" in the brackets is for 1st section rather than last start.

```
HrSectionTime( ,1)
```

Or

```
HrSectionTime(1,1)
```

The above 2 functions return the same value, i.e., time (in seconds) for the first section of the race of the last start. Please note that the last "1" in the brackets refers to last start.

```
HrSectionTime(2)
```

Returns time for the 2nd section of the race.

**See also**

HrSectionalTime( ), JrSectionalTime( ), SrSectionalTime( ) | HrAccSectionalTime( ), JrAccSectionalTime( ), SrAccSectionalTime( )

## 16.6.47 HrSectionTimew( ), JrSectionTimew( ), SrSectionTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrSectionTime( ), JrSectionTime( ), SrSectionTime( )

**Returns**

Numeric (Width=5, Decimal=2)

## 16.6.48 HrPowerFigure( ), JrPowerFigure( ), SrPowerFigure( )

The functions return the speed figure on the Speed Power page of the Hong Kong Jockey Club website. In Macau edition, these functions return blank values.

**Syntax**

```
HrSpeedFigure(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrSpeedFigure(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrSpeedFigure(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric (Width=3)

**Example of Return Value**

101

© Welkin Systems Limited

**See also**

HrPowerComments( ), JrPowerComments( ), SrPowerComments( )

## 16.6.49 HrPowerComments( ), JrPowerComments( ), SrPowerComments( )

The functions return the comments on horses on the Speed Power page of the Hong Kong Jockey Club website. In Macau edition, these functions return blank values.

**Syntax**

```
HrPowerComments(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrPowerComments(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrPowerComments(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Multiline Text

**See also**

HrPowerFigure( ), JrPowerFigure( ), SrPowerFigure( )

## 16.6.50 HrTrack( ), JrTrack( ), SrTrack( )

The functions return the race track together with the width adjustment of the track of the race. For example, for an "A+3" track, the functions return a numeric value "A+3".

**Syntax**

```
HrTrack(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrTrack(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrTrack(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=8)

**Example of Return Value**

"C+3 "

**See also**

HrTrackID( ), JrTrackID( ), SrTrackID( ) | HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( ) | HrTrackShift( ), JrTrackShift( ), SrTrackShift( )

## 16.6.51 HrTrackID( ), JrTrackID( ), SrTrackID( )

The functions return a character identifying the track of the race. Here are some possible values:

| ID | Description |
|---|---|

© Welkin Systems Limited

| | |
|---|---|
| A | Track A |
| B | Track B |
| C | Track C |
| 1 | New A |
| 2 | New B |
| 3 | New C |
| W | All Weather Track |

**Syntax**

```
HrTrackID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrTrackID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrTrackID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=1)

**Example of Returned Value**

"A"

**See also**

HrTrack( ), JrTrack( ), SrTrack( ) | HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( ) | HrTrackShift( ), JrTrackShift( ), SrTrackShift( )

## 16.6.52 HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( )

The functions return the width adjustment of the track of the race. For example in a character string.

**Syntax**

```
HrTrackPlus(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrTrackPlus(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrTrackPlus(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=2)

**Example of Returned Value**

"+3"

**See also**

HrTrack( ), JrTrack( ), SrTrack( ) | HrTrackID( ), JrTrackID( ), SrTrackID( ) | HrTrackShift( ), JrTrackShift( ), SrTrackShift( )

© Welkin Systems Limited

## 16.6.53 HrTrackShift( ), JrTrackShift( ), SrTrackShift( )

The functions return the width adjustment of the track of the race. For example, for a "A+3" track, the functions return a numeric value 3.

### Syntax

```
HrTrackShift(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrTrackShift(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrTrackShift(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Numeric (Width=2)

### Example of Returned Value

3

### See also

HrTrack( ), JrTrack( ), SrTrack( ) | HrTrackID( ), JrTrackID( ), SrTrackID( ) | HrTrackPlus( ), JrTrackPlus( ), SrTrackPlus( )

## 16.6.54 HrWinner( ), JrWinner( ), SrWinner( )

The functions return the name of the winning horse in the race of the horse.

### Syntax

```
HrWinner(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrWinner(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrWinner(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=8 in Chinese Mode, Width=30 in English Mode)

## 16.6.55 HrWinnerID( ), JrWinnerID( ), SrWinnerID( )

The functions return the internal ID of the winning horse in the race of the horse.

### Syntax

```
HrWinnerID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrWinnerID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrWinnerID(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

### Returns

Character (Width=3)

### Example of Returned Value

© Welkin Systems Limited

"0AX"

### 16.6.56 HrWinSeconds( ), JrWinSeconds( ), SrWinSeconds( )

The functions return number of seconds the winner took to finish the run.

**Syntax**

```
HrWinSeconds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrWinSeconds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrWinSeconds(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Numeric

**Example of Returned Value**

107.4

### 16.6.57 HrWinTime( ), JrWinTime( ), SrWinTime( )

The functions return the finishing time of the winner of the race of the horse.

**Syntax**

```
HrWinTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
JrWinTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
SrWinTime(nBackRaces,'fCondition',cDefaultVal,cIncluding)
```

**Returns**

Character (Width=6)

**Example of Returned Value**

"1.23.3"

### 16.6.58 HrWinTimew( ), JrWinTimew( ), SrWinTimew( )

These functions are identical to their original versions, except that they return 2 digits after the decimal point. Click on the link below to see the syntax of the functions.

HrWinTime( ), JrWinTime( ), SrWinTime( )

**Returns**

Character (Width=7)

© Welkin Systems Limited

### 16.6.59 IrregularRace( )

If the current race is not an ordinary Hong Kong race, the function returns false (.F.), otherwise (e.g., an international race) the function returns true (.T.).

**Syntax**

IrregularRace( )

**Returns**

Logical

## 16.7 Data Collection Assistant Functions

The functions in this chapter do not return a particular race datum by itself. Most of them return the data collected by other MCL functions. Some functions help other MCL functions confirm the statuses or positions in views. We called this group of functions "data collection assistant functions".

**>>> Subtopics of this section <<<**

- AtStart( )
- Column( )
- CurRun( )
- GetColumn( )
- HrData( ), JrData( ), SrData( )
- HrHistoryList( ), JrHistoryList( ), SrHistoryList( )
- IfAtTop( )
- IfCurRun( )
- RaceVar( )
- StartVar( )

### 16.7.1 AtStart( )

This function returns the result of an MCL expression at a particular start of the horse. The 2nd argument of the function requires a Start ID.

**Syntax**

```
AtStart('eMCLExpr',cStartID)
```

**Returns**

The data type of the return value is identical to that returned by eMCLExpr.

**Example**

Create a column "FastestStartID" as follows:

```
HrMinStartID('HrTime( )',,'For HrDistance( )='+Str(HrDistance( )))
```

The column returns the ID of the start, among all starts of distance same as the current race, at which the MCL expression HrTime( ) is a minimum. The returned value

© Welkin Systems Limited

may be something like: "1W820015".

Then, you may get the weight, rating, and FP of the horse of that start by putting the following functions in subsequent columns as follows:

```
AtStart('HrWeight( )',Column('FastestStartID'))
AtStart('HrRating( )',Column('FastestStartID'))
AtStart('HrFP( )',Column('FastestStartID'))
```

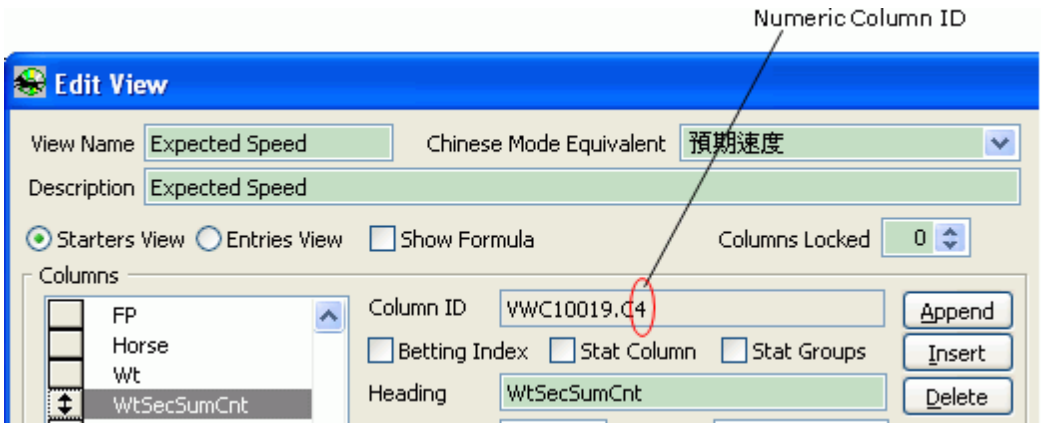
## 16.7.2 Column( )

The Column( ) function returns the value of a column of a race view.

### Syntax

1. Column(cColumnHeading)
2. Column(nNumericColumnID)

| Argument | Description |
|---|---|
| cColumnHeading | Heading caption of column<br>Important: The argument cColumnHeading is case-sensitive. You must use Column('FP'), not Column('Fp'), to retrieve data of a column of name 'FP'. |
| nNumericColumnID | Numeric column ID is the rightmost number in the full Column ID. The following figure shows how to find the Column ID in the View Definition Editing Form |




### Returns

Value returned by the function will be of the same type and width of the referencing column.

### Examples

This following expression returns average of the columns This Rating and Last Rating.

```
(Column('This Rating')+Column('Last Rating'))/2
```

If the numeric column ID of the This Rating column and the Last Rating column are 3 and 5 respectively, the above expression may be rewritten as:

© Welkin Systems Limited

```
(Column(3)+Column(5))/2
```

**See also**

GetColumn( )

### 16.7.3 CurRun( )

This function returns the result of an MCL expression at the viewing start of the horse. To some extend, this function is similar to the AtStart( ) function with a default starter ID of the underlying race, i.e., the view that you are viewing. It is important to note the following limitations:

1. CurRun( ) function will return .NULL. in model evaluation since "viewing race" will be undefined when you evaluate a model (clicking the [Eval] button in the Race Meeting form).
2. Since viewing race changes whenever you jump to a new race, cached old values returned by CurRun( ) will be out-of-date all the time. You are recommended to set the column type to "No Caching", otherwise you will need to push the [Recalc] button to update the values whenever a new race is displayed.

**Syntax**

```
CurRun('eMCLExpr')
```

**Returns**

The data type of the return value is identical to that returned by eMCLExpr.

**Example**

Create a column "WeightDifference" as follows:

```
HrWeight( )-AtViewingRace('HrWeight( )')
```

The column returns different between the weight in each past start and the weight in the current start of each horse.

### 16.7.4 GetColumn( )

This function imports column values of another view of the current race into the current column of the current view. The column to be imported can be either a calculated or an input column. The system will import the column values from RaceMate's view data cache. Please note the following points about using the function:

1. The target race view, from which the data to be imported, must be opened at least once (to load the data into the view cache) before the current view, running the GetColumn( ) function, is opened. Otherwise the GetColumn( ) function will return .NULL. (Doing statistics and model evaluations will also populate data into the view cache.)
2. If you choose "Clean Up Calculation Results in Cache" from the main menu, all calculated column data in the view cache will be lost. All the GetColumn( ) functions that import values from calculated columns will return .NULL. You have to

© Welkin Systems Limited

repopulate the view cache in order to supply necessary data for the function. However, data of input columns will not be clear by the "Clean Up Calculation Results in Cache" menu option, therefore getting values from input columns will not be affected.

#### Syntax

```
GetColumn(cColumnID)
```

#### Argument

cColumnID is the Column ID of the column of the target view. You can see the Column ID of a column on the Edit View form (see figure below).

#### Returns

Value returned by the function will be of the same type and width of the target column.

#### Examples

```
GetColumn('VWC10017.C6')
```

Retrieve the values of the Yeung's Index column (Column ID='VWC10017.C6') of the Young's Index view into the current column of the current view.

#### See also

Column( )

### 16.7.5 HrData( ), JrData( ), SrData( )

The functions return the value of an MCL expression, which is passed into the function through the first argument.

#### Syntax

```
HrData(cMCLExpr,nBackRaces,'fCondition',nDefaultVal)
JrData(cMCLExpr,nBackRaces,'fCondition',nDefaultVal)
SrData(cMCLExpr,nBackRaces,'fCondition',nDefaultVal)
```

#### Returns

The same data type as MCL expression passed into the first argument.

#### Example

```
HrData('HrWeight( )/HrBodyweight( )',1)
```

This function calculates the Loading-Bodyweight ratio of a horse in last race. Although the function can be rewritten as HrWeight(1)/HrBodyweight(1) and identical results will be returned, however, the speed of latter expression will be lower than the example. It is because in the latter expression HrWeight( ) and HrBodyweight( ) will each walk though the past records once to retrieve corresponding information respectively while the HrData( ) function of the example walks through the past record only once. When

© Welkin Systems Limited

the condition ( fCondition ) getting a target race is complicated, the speed improvement of the HrData( ) function will be obvious.

### 16.7.6 HrHistoryList( ), JrHistoryList( ), SrHistoryList( )

The functions return a string composed of a list of items got by concatenating the values of an MCL expression in the past records of a horse. The values will be separated by commas.

#### Syntax

```
HrHistoryList('cValueExpr',nRaces,'fCondition' ,cIncluding,cSeparator,lRecentFirst)
JrHistoryList('cValueExpr',nRaces,'fCondition' ,cIncluding,cSeparator,lRecentFirst)
SrHistoryList('cValueExpr',nRaces,'fCondition' ,cIncluding,cSeparator,lRecentFirst)
```

| Argument | Description | Example |
|---|---|---|
| 'cValueExpr' | • Expression returning the target datum.<br>• lCountIf can be of any data type. If lCountIf is of logical type, all true (.T.) values will be counted. For any other types, record will be counted only if lCountIf returns a non-null value. | 'HrFP( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 4 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For '+Same(HrDistance( )) |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |
| cSeparator | Separator, which is default to "," | "-" |

© Welkin Systems Limited

| | | |
|---|---|---|
| lRecentFirst | If .T., lastest items will be put first. (Recent items are put to the right by default) | .T. |

**Returns**

Character

**Example**

```
HrHistoryList('HrFP( )',4)
```

Finishing positions of last 4 starts. (Example returned string: "2, 3, 9, 1")

```
HrHistoryList('HrFP( )',4,'For '+Same('HrDistance( )'))
```

Finishing positions of last 4 starts of distance same as the current start. (Example returned string: "2, 3, 9, 1")

```
HrHistoryList('HrFP( )',4,'For '+Same('HrDistance( )'),,'-',.T.)
```

Finishing positions of last 4 starts of distance same as the current start. Latest finishing positions will be put first (Example returned string: "2, 3, 9, 1")

## 16.7.7 IfAtTop( )

For the horse at the top row of a starters view in natural order, the function will return the value of the expression embedded in the string of the first argument. For horses at other rows, the function will return the value of the expression embedded in string of the second argument (.NULL. will be returned if the second argument is missing). We can use this function to get a datum from one and only one horse of a race. Incorporating with the statistics (pushing the 'Stat' button) feature, we can calculate statistics on a race-based (instead of horse based) item.

"Natural order" in the above paragraph means:

- In past races: ordered by finishing positions
- In races without finishing positions: ordered by horse numbers

**Syntax**

```
IfAtTop('fTrueExpression','fFalseExpression')
```

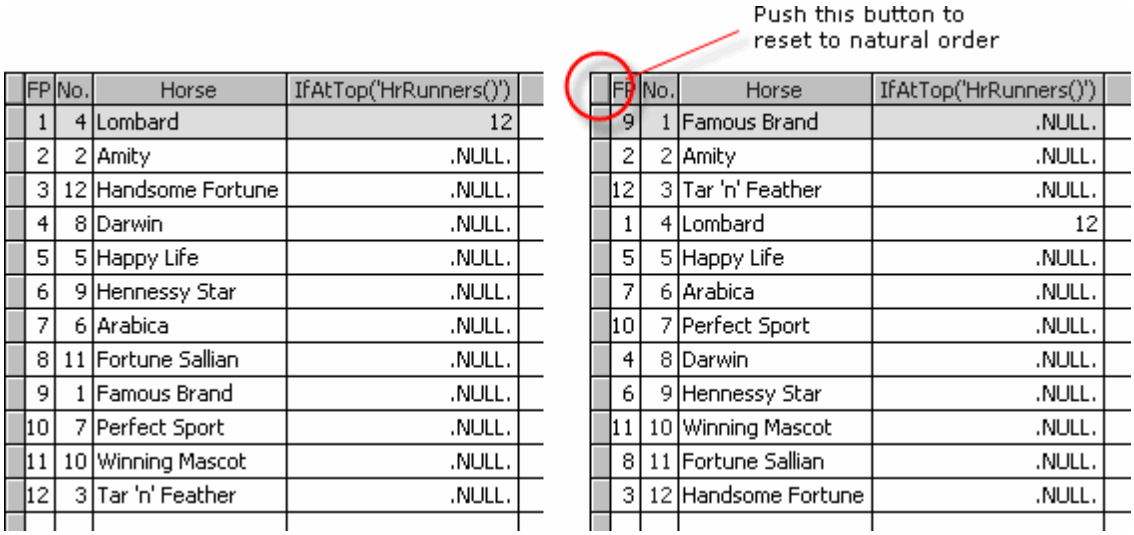
**Returns**

The same data type as the expression contained in the argument strings

**Example**

In the following example, the formula IfAtTop('HrRunners( )') is applied to the fourth column of the view. Lombard is the horse at top of the view in natural order. No matter what current order the view is in (e.g., in the right figure the view is sorted by horse names), the function will always return 12 (number of runners) in the row of Lombard and return .NULL. in other rows.

© Welkin Systems Limited




| FP | No. | Horse | IfAtTop('HrRunners()') |
|---|---|---|---|
| 1 | 4 | Lombard | 12 |
| 2 | 2 | Amity | .NULL. |
| 3 | 12 | Handsome Fortune | .NULL. |
| 4 | 8 | Darwin | .NULL. |
| 5 | 5 | Happy Life | .NULL. |
| 6 | 9 | Hennessy Star | .NULL. |
| 7 | 6 | Arabica | .NULL. |
| 8 | 11 | Fortune Sallian | .NULL. |
| 9 | 1 | Famous Brand | .NULL. |
| 10 | 7 | Perfect Sport | .NULL. |
| 11 | 10 | Winning Mascot | .NULL. |
| 12 | 3 | Tar 'n' Feather | .NULL. |

| FP | No. | Horse | IfAtTop('HrRunners()') |
|---|---|---|---|
| 9 | 1 | Famous Brand | .NULL. |
| 2 | 2 | Amity | .NULL. |
| 12 | 3 | Tar 'n' Feather | .NULL. |
| 1 | 4 | Lombard | 12 |
| 5 | 5 | Happy Life | .NULL. |
| 7 | 6 | Arabica | .NULL. |
| 10 | 7 | Perfect Sport | .NULL. |
| 4 | 8 | Darwin | .NULL. |
| 6 | 9 | Hennessy Star | .NULL. |
| 11 | 10 | Winning Mascot | .NULL. |
| 8 | 11 | Fortune Sallian | .NULL. |
| 3 | 12 | Handsome Fortune | .NULL. |

Remark: No matter what current order is, when you push the left top small button of the grid, the view will be reset to the natural order.

## 16.7.8 IfCurRun( )

In the current run row of a view, the function will return the value of the expression inside the character string of the first argument. In past record rows, the function will return the value of the expression inside the character string of the second argument.

### Syntax

```
IfCurRun('fTrueExpression','fFalseExpression')
```

### Returns

The same data type as the expression contained in the argument strings

### Example

In the view shown in the following figure, the formula CurRunHorses( ), in the fourth column, shows the current starters met by each horse in each race (including the current race). The formula IfCurRun([" "],[CurRunHorses( )]) applies the CurRunHorses ( ) function only in past races. A space (" ") will be shown in the current run.

© Welkin Systems Limited

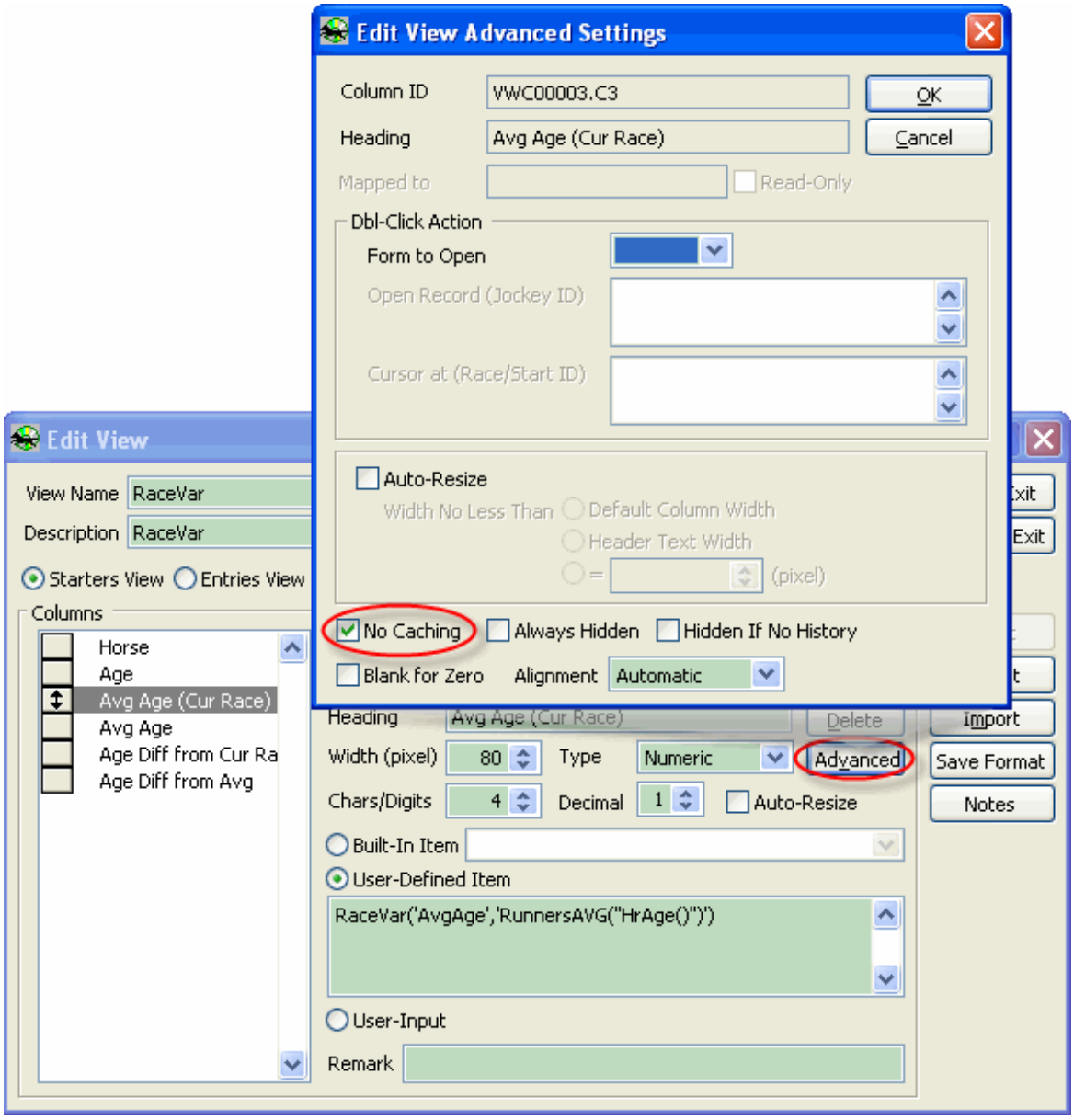
| | FP | No | Horse | CurRunHorses() | IfCurRunFn([" "],[CurRunHorses()]) |
|---|---|---|---|---|---|
| | 1 | 10 | Luckswell | High Spirits,Rebel Leader,Good Fit,Danefactor,D | |
| | 5 | 8 | 01-152 | Don Sebastian,Win For Lily,High Sky,Viper Corp | Don Sebastian,Win For Lily,High Sky,Viper Corp |
| | 2 | 10 | 01-100 | Danefactor,Viper Corp | Danefactor,Viper Corp |
| | 10 | 13 | 01-057 | Win For Lily,Danefactor | Win For Lily,Danefactor |
| | 2 | 3 | High Spirits | Luckswell,Rebel Leader,Good Fit,Danefactor,Do | |
| | 9 | 11 | 01-175 | | |
| | 9 | 2 | 01-116 | Rebel Leader,Don Sebastian,Viper Corp | Rebel Leader,Don Sebastian,Viper Corp |
| | 6 | 11 | 01-082 | | |
| | 3 | 4 | Rebel Leader | Luckswell,High Spirits,Good Fit,Danefactor,Don | |
| | 2 | 10 | 01-158 | Danefactor,My Heart | Danefactor,My Heart |
| | 2 | 7 | 01-116 | High Spirits,Don Sebastian,Viper Corp | High Spirits,Don Sebastian,Viper Corp |
| | 3 | 8 | 01-072 | Danefactor,High Sky,Viper Corp | Danefactor,High Sky,Viper Corp |
| | 4 | 13 | Good Fit | Luckswell,High Spirits,Rebel Leader,Danefactor, | |
| | 13 | 3 | 01-107 | | |

### 16.7.9 RaceVar( )

The function stores the result of an MCL expression under a name. We call the name a race variable. When you use the function RaceVar(cVariableName,"eMCLExpr") in a row of a view of a race, the system will see if the variable has already been defined in this race. If the variable has already been defined, the function will return the value stored in the variable. Otherwise it will evaluate the expression eMCLExpr and store the result into the variable (the variable will automatically be created) and return the result. Please note the following points about a race variable:

1. The value of a race variable of a race is independent of the same variable for another race in the same view. The expression eMCLExpr will be evaluated once for each race.
2. When you expend a view to show the past records, although the lines are not showing the data of the current race, the race variable will not be recalculated.
3. Whenever the formula for a view column contains a RaceVar( ) function, the system will automatically set the column as "No Caching" (figure below) so that values in the view data cache (particularly those put into the past records of an expanded view) will be refreshed in other races.

© Welkin Systems Limited




4. The values of race variables are store in a variable cache that will not be refreshed when you push the "Recalc" key. To refresh the variable cache, you have to hold down the Shift key when you click on the "Recalc" key.
5. Start Variables (saved and accessed via RaceVar( )) and Race Variables (saved and accessed via StartVar( )) are store in different area of the variable cache. They will not interference with each other even the same variable name is given to both a Race Variable and a Start Variable.

### Syntax

`RaceVar(cVariableName,"eMCLExpr")`

### Returns

Data type returned by the expression eMCLExpr

© Welkin Systems Limited

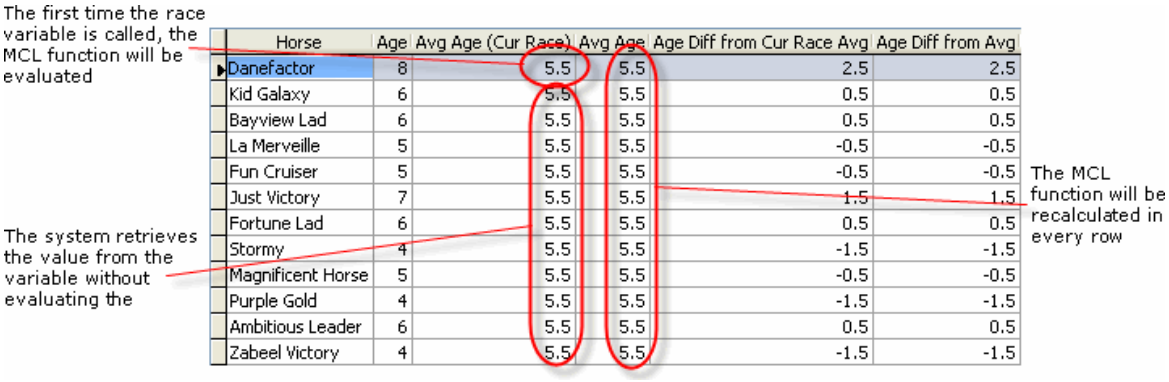
### Example

```
RaceVar('AvgAge','RunnersAVG("HrAge( )")')
```

The function puts the average age of the current starters into the race variable AvgAge. Functions in all view lines can then get access to this value.

The structure of example view in the following figure is as follows:

| Column | Formula | Type |
|---|---|---|
| Horse | (Built-in column) | |
| Age | (Built-in column) | |
| Avg Age (Cur Race) | RaceVar('AvgAge','RunnersAVG("HrAge( )")') | N(4,1) |
| Avg Age | RunnersAVG("HrAge( )") | N(4,1) |
| Age Diff from Cur Race Avg | HrAge( )-Column('Avg Age (Cur Race)') | N(4,1) |
| Age Diff from Avg | HrAge( )-Column('Avg Age') | N(4,1) |


The first time the race variable is called, the MCL function will be evaluated
The system retrieves the value from the variable without evaluating the
The MCL function will be recalculated in every row

| Horse | Age | Avg Age (Cur Race) | Avg Age | Age Diff from Cur Race Avg | Age Diff from Avg |
|---|---|---|---|---|---|
| Danefactor | 8 | 5.5 | 5.5 | 2.5 | 2.5 |
| Kid Galaxy | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| Bayview Lad | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| La Merveille | 5 | 5.5 | 5.5 | -0.5 | -0.5 |
| Fun Cruiser | 5 | 5.5 | 5.5 | -0.5 | -0.5 |
| Just Victory | 7 | 5.5 | 5.5 | 1.5 | 1.5 |
| Fortune Lad | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| Stormy | 4 | 5.5 | 5.5 | -1.5 | -1.5 |
| Magnificent Horse | 5 | 5.5 | 5.5 | -0.5 | -0.5 |
| Purple Gold | 4 | 5.5 | 5.5 | -1.5 | -1.5 |
| Ambitious Leader | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| Zabeel Victory | 4 | 5.5 | 5.5 | -1.5 | -1.5 |

In the view of the above figure, the numbers in the columns "Avg Age (Cur Race)" and "Avg Age" are the same. However, the MCL function in the "Avg Age" column will be recalculated in every row while the MCL function inside the RaceVar( ) function of the "Avg Age (Cur Race)" column will only be calculated only in the first row. Therefore, the speed of showing the latter column is much faster.

When the view is expanded to show the past records as show in the following figure, the numbers in the past records of the "Avg Age (Cur Race)" column are got from the variable AvgAge without recalculation. Therefore these figures are different from those in the "Avg Age" column.

| Horse | Age | Avg Age (Cur Race) | Avg Age | Age Diff from Cur Race Avg | Age Diff from Avg |
|---|---|---|---|---|---|
| Danefactor | 8 | 5.5 | 5.5 | 2.5 | 2.5 |
| 05-047 | 8 | 5.5 | 5.4 | 2.5 | 2.6 |
| 04-611 | 0 | 5.5 | 4.7 | -5.5 | -4.7 |
| Kid Galaxy | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| 05-044 | 6 | 5.5 | 5.6 | 0.5 | 0.4 |
| 05-027 | 6 | 5.5 | 5.3 | 0.5 | 0.7 |
| Bayview Lad | 6 | 5.5 | 5.5 | 0.5 | 0.5 |
| 05-035 | 6 | 5.5 | 5.0 | 0.5 | 1.0 |
| 05-020 | 6 | 5.5 | 5.3 | 0.5 | 0.7 |
| La Merveille | 5 | 5.5 | 5.5 | -0.5 | -0.5 |

### See also

StartVar( )

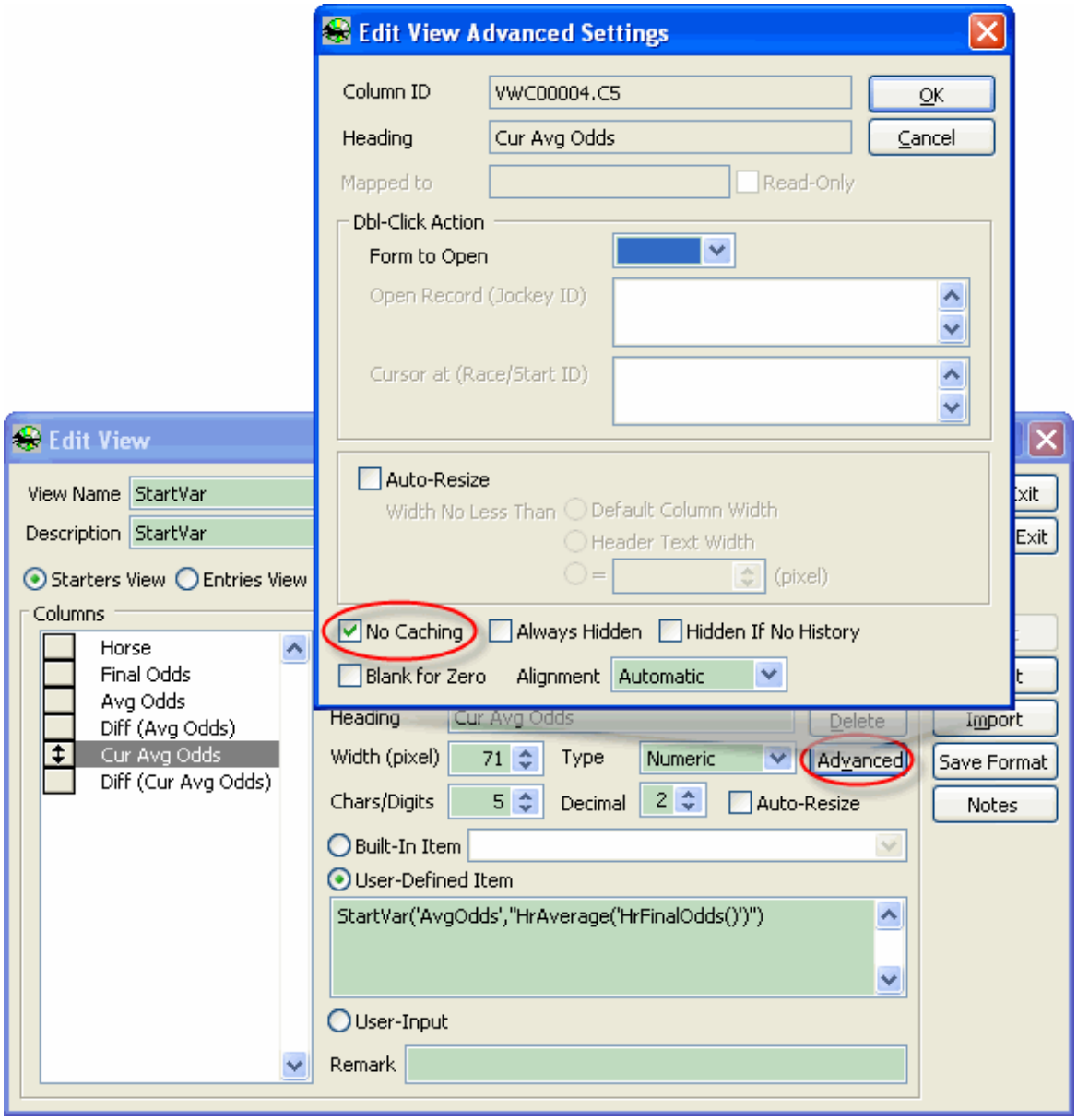
© Welkin Systems Limited

### 16.7.10 StartVar( )

This function evaluates an MCL expression and stores the result into a variable, namely Start Variable. When the function StartVart(cVariableName, "eMCLExpr") is executed in a horse row of a race in a view, the system will check if the variable is already defined for the horse in the race. If the variable has already been defined for this start, the function will return the value stored in the variable. Otherwise it will evaluate the MCL expression eMCLExpr and store the result into the variable (the variable will automatically be created) and return the result. Please note the following points about race variables:

1. The value of a starter variable for a horse is independent of the same variable for another horse of the current or another race. The expression eMCLExpr will be evaluated for each horse of each race.
2. When you expend a view to show the past records, although the lines are not showing the data of the current race, the starter variable will not be recalculated.
3. Whenever the formula for a view column contains a StartVar( ) function, the system will automatically set the column type as "No Caching" (figure below) so that values in the view data cache (particularly those put into the past records of an expanded a view) will be refreshed automatically in other races.

© Welkin Systems Limited




4. The values of race variables are store in a variable cache that will not be refreshed when you push the "Recalc" key. To refresh the variable cache, you have to hold down the Shift key when you click on the "Recalc" key.
5. Start Variables (saved and accessed via RaceVar( )) and Race Variables (saved and accessed via StartVar( )) are store in different area of the variable cache. They will not interference with each other even the same variable name is given to both a Race Variable and a Start Variable.

### Syntax

`StartVar(cVariableName,"eMCLExpr")`

### Returns

Retruns Data type returned by the expression eMCLExpr

© Welkin Systems Limited

**Example**

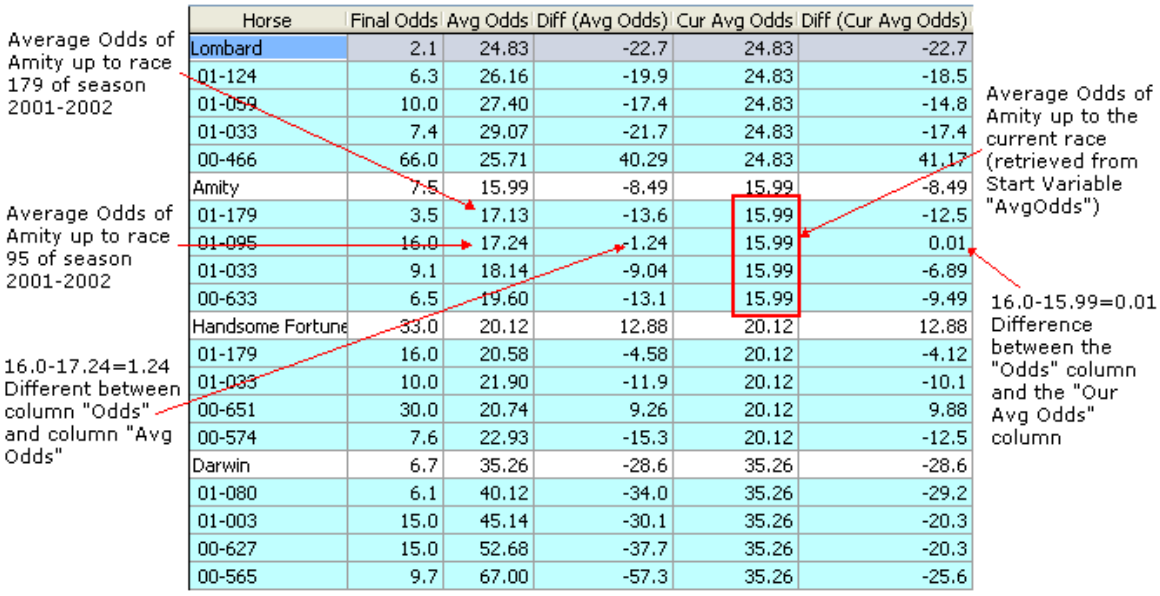
```
StartVar('AvgTime',"HrAverage('HrSeconds( )', ,Same('HrDistance( )'))")
```

When the function is firstly called in a horse row of a view, it stores the average time, in seconds, of the horse into the race variable AvgTime. Subsequent calls to the function for the same horse will return the value directly from the variable AvgTime.

The example view in the figure below shows the 221st race of season 2001-2002. The structure of the view is as follows:

| Column | Formula | Width |
|---|---|---|
| Horse | (Built-in column : Horse Name) | |
| Odds | (Built-in column : Final Odds) | |
| Avg Odds | HrAverage('HrFinalOdds( )') | N(5,2) |
| Diff (Avg Odds) | Column('Odds')-Column('Avg Odds') | N(5,2) |
| Cur Avg Odds | StartVar('AvgOdds',"HrAverage('HrFinalOdds ( )')") | N(5,2) |
| Diff (Cur Avg Odds) | Column('Odds')-Column('Cur Avg Odds') | N(5,2) |

In this example view, the function of the "Cur Avg Odds" column stores the average odds up to the current race into the Start Variable "AvgOdds", in the past record lines of the view, the same function retrieves the value directly from the variable.


Average Odds of Amity up to race 179 of season 2001-2002
Average Odds of Amity up to race 95 of season 2001-2002
16.0-17.24=1.24 Different between column "Odds" and column "Avg Odds"
Average Odds of Amity up to the current race (retrieved from Start Variable "AvgOdds")
16.0-15.99=0.01 Difference between the "Odds" column and the "Our Avg Odds" column

| Horse | Final Odds | Avg Odds | Diff (Avg Odds) | Cur Avg Odds | Diff (Cur Avg Odds) |
|---|---|---|---|---|---|
| Lombard | 2.1 | 24.83 | -22.7 | 24.83 | -22.7 |
| 01-124 | 6.3 | 26.16 | -19.9 | 24.83 | -18.5 |
| 01-059 | 10.0 | 27.40 | -17.4 | 24.83 | -14.8 |
| 01-033 | 7.4 | 29.07 | -21.7 | 24.83 | -17.4 |
| 00-466 | 66.0 | 25.71 | 40.29 | 24.83 | 41.17 |
| Amity | 7.5 | 15.99 | -8.49 | 15.99 | -8.49 |
| 01-179 | 3.5 | 17.13 | -13.6 | 15.99 | -12.5 |
| 01-095 | 16.0 | 17.24 | -1.24 | 15.99 | 0.01 |
| 01-033 | 9.1 | 18.14 | -9.04 | 15.99 | -6.89 |
| 00-633 | 6.5 | 19.60 | -13.1 | 15.99 | -9.49 |
| Handsome Fortune | 33.0 | 20.12 | 12.88 | 20.12 | 12.88 |
| 01-179 | 16.0 | 20.58 | -4.58 | 20.12 | -4.12 |
| 01-033 | 10.0 | 21.90 | -11.9 | 20.12 | -10.1 |
| 00-651 | 30.0 | 20.74 | 9.26 | 20.12 | 9.88 |
| 00-574 | 7.6 | 22.93 | -15.3 | 20.12 | -12.5 |
| Darwin | 6.7 | 35.26 | -28.6 | 35.26 | -28.6 |
| 01-080 | 6.1 | 40.12 | -34.0 | 35.26 | -29.2 |
| 01-003 | 15.0 | 45.14 | -30.1 | 35.26 | -20.3 |
| 00-627 | 15.0 | 52.68 | -37.7 | 35.26 | -20.3 |
| 00-565 | 9.7 | 67.00 | -57.3 | 35.26 | -25.6 |

**See Also**

RaceVar( )

## 16.8 Statistics on Past Records

This group of functions returns statistical figures on the past records of horses, jockeys, and stables.

**>>> Subtopics of this section <<<**

© Welkin Systems Limited

- HrAverage( ), JrAverage( ), SrAverage( )
- HrAverageBy( ), JrAverageBy( ), SrAverageBy( )
- HrAverageNZ( ), JrAverageNZ( ), SrAverageNZ( )
- HrCount( ), JrCount( ), SrCount( )
- HrCountBy( ), JrCountBy( ), SrCountBy( )
- HrCountNZ( ), JrCountNZ( ), SrCountNZ( )
- HrMaximum( ), JrMaximum( ), SrMaxmum( )
- HrMaxStartID( ), JrMaxStartID( ), SrMaxStartID( )
- HrMinimum( ), JrMinimum( ), SrMinimum( )
- HrMinStartID( ), JrMinStartID( ), SrMinStartID( )
- HrStatList( ), JrStatList( ), SrStatList( )
- HrSum( ), JrSum( ), SrSum( )
- HrSumBy( ), JrSumBy( ), SrSumBy( )
- HrSumCount( ), JrSumCount( ), SrSumCount( )
- HrXAverage( ), JrXAverage( ), SrXAverage( )
- HrXMaximum( ), JrXMaximum( ), SrXMaximum( )
- HrXMaxStartID( ), JrXMaxStartID( ), SrXMaxStartID( )
- HrXMinimum( ), JrXMinimum( ), SrXMinimum( )
- HrXMinStartID( ), JrXMinStartID( ), SrXMinStartID( )
- HrXSum( ), JrXSum( ), SrXSum( )
- StdTime( )
- StdTimew( )

### 16.8.1 HrAverage( ), JrAverage( ), SrAverage( )

The functions return averages of specified information in the past records of the horse, jockey, or stable respectively.

#### Syntax

```
HrAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
JrAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
SrAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the average of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Numeric

**Examples**

```
HrAverage('HrWeight( )',3)
```

The above example returns the average loading of the past 3 races of the horse.

```
SrAverage('HrSpeed( )',4,'For HrDistance( )=1400')
```

The above example returns the average speed of the last 4 1400m starts of the stable.

```
JrAverage('HrSpeed( )',,'For HrDistance( )=1400 and HrAveRating( )>100',95)
```

The above example returns the average speed of all runs meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

If no race meeting the condition can be found, return 95.

```
SrAverage('HrFP( )',10,,'R')
```

The above function returns the average of the finishing position of last 10 starts including the starts in the underlying race.

## 16.8.2 HrAverageBy( ), JrAverageBy( ), SrAverageBy( )

The functions return the average of a datum over the past records, which satisfy a specific condition, of the horse, jockey and stable respectively. The results will be broken down by the value of an expression and returned as a character string.

© Welkin Systems Limited

### Syntax

```
HrAverageBy('nValueExpr','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrAverageBy('nValueExpr','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrAverageBy('nValueExpr','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

### Enhanced Syntax

By using the enhanced syntax you may:

1. carry out multiple statistics in a single pass of past record scanning
2. customize the format of return string
3. define the order of breakdown groups
4. define a summary session in the return string

The enhanced syntax of these functions is similar to that for HrCountBy( ), JrCountBy ( ), SrCountBy( ). Please see its help page for details.

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the average of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| 'eBreakdownExpr' | An expression by which results will be broken down. Omitting this argument tells the function do the statistics over all legitimate records. | 'HrDistance( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return | |

© Welkin Systems Limited

| | | |
|---|---|---|
| | value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character (The length of the returned character string depends on the breakdown expression. It is recommended to set the column data type to Multiline Text, so that number of characters returned can be of any length.)

**Examples**

```
HrAverageBy('HrFp( )','HrDistance( )')
```

The above function returns the average finishing position of a horse over all races. The result will then be broken down by distance.

```
HrAverageBy('HrFp( )','HrDistance( )',, 'HrFp( )=1')
```

The above function returns the average finishing position of a horse over all winning races. The result will then be broken down by distance.

**See also**

HrCountBy( ), JrCountBy( ), SrCountBy( ) | HrSumBy( ), JrSumBy( ), SrSumBy( ) | HrAverage( ), JrAverage( ), SrAverage( )

### 16.8.3 HrAverageNZ( ), JrAverageNZ( ), SrAverageNZ( )

This set of functions is similar to the HrAverage( ) function set. However, these functions will skip items with values 0 and Null. (NZ stands No Zero)

**Syntax**

```
HrAverageNZ('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
JrAverageNZ('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
SrAverageNZ('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the average of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| | function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Numeric

**See Also**

HrAverage( ), JrAverage( ), SrAverage( ) | HrAverageBy( ), JrAverageBy( ), SrAverageBy( )

## 16.8.4 HrCount( ), JrCount( ), SrCount( )

The functions return number of runs that meet the specified condition in the past records of the horse, jockey and stable respectively.

**Syntax**

```
HrCount('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
JrCount('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
SrCount('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'lCountIf' | • Datum the count of which is to be found. | 'HrFP( )=1' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | • lCountIf can be of any data type. If lCountIf is of logical type, all true (.T.) values will be counted. For any other types, record will be counted only if lCountIf returns a non-null value. | |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Numeric

**Examples**

```
HrCount('HrRating( )>80',6,'For HrDistance( )=1400')
```

The above example counts in last six 1400m starts in which the bodyweight of the horse is greater than 1000. Since the first argument of the function is a logical expression, RaceMate will count the starts (which satisfy the condition in the 3rd argument) if the expression returns true.

```
HrCount('HrRating( )',6,'For HrDistance( )=1400')
```

© Welkin Systems Limited

The above example returns the number of 1400m starts, in which HrRating( ) returns a non-null value, in the past record of the horse. If total number of races is greater than 6, the function will return 6 instead of the actual number. Please note that starts will be counted even if HrRating( ) returns 0.

```
JrCount(,'While JrSeason( )='+Str(JrSeason( ),4))
```

The above example returns the number of horses that the jockey has ride in this season.

```
HrCount(,'For HrDistance( )=1400 and HrAveRating( )>100')
```

The above example returns the number of races meeting the following conditions in the past records.

1. 1400m race
2. Average rating of the race is greater than 100

**See also**

HrCountNZ( ), JrCountNZ( ), SrCountNZ( )

## 16.8.5 HrCountBy( ), JrCountBy( ), SrCountBy( )

The functions count the number of runs that meet a specific criterion in the past records of a horse, jockey and stable respectively. The counts will be broken down by an expression value and returned as a character string.

**Syntax**

```
HrCountBy('lCountIf','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrCountBy('lCountIf','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrCountBy('lCountIf','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

**Enhanced Syntax**

Arguments 'lCountIf' and 'eBreakdownExpr' support enhanced syntax, which will be explained in the lower half of this page. By using the enhanced syntax you may:

- carry out multiple statistics in a single pass of past record scanning
- customize the format of return string
- change the order of breakdown groups
- define a summary session in the return string
- define a string to separate statistical results

| Argument | Description | Example |
|---|---|---|
| 'lCountIf' | • Datum the count of which is to be found.<br>• lCountIf can be of any data type. If lCountIf is of logical type, all true (.T.) values will be counted. For any other types, record will be counted only if lCountIf returns a non-null value. | 'HrFP( )=1' |
| 'eBreakdownExpr' | An expression by which results will be broken down. Omitting this argument | 'HrDistance( )' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | tells the function do the statistics over all legitimate records. | |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character (The length of the returned character string depends on the breakdown expression. It is recommended to set the column data type to Multiline Text, so that the number of characters returned can be of any length.)

**Examples**

```
HrCountBy( ,'HrDistance( )')
```

The above function counts the number of races of each horse broken down by distance.

```
HrCountBy('HrFp( )=1','HrDistance( )')
```

The above function counts the number of winning races of each horse broken down by distance.

© Welkin Systems Limited

```
HrCountBy( ,'HrDistance( )', ,'For HrFp( )=1')
```

The above function counts the number of races in which the horse wins. The result is then broken down by distance.

The following picture shows the difference among the examples above:

- In the first 2 examples, no filtering condition (4th argument) is passed into the MCL function. Therefore all races will be processed.
- In the 1st example all races, no matter winning or losing, will be counted. In the 2nd example, only records with HrFp( )=1 will be counted. All distances will be returned in the resulting string.
- In the third example, only the races satisfying the filtering condition ('For HrFp( )=1') will be processed. Therefore all losing races (HrFp( )=1 is not true) will neither be processed nor found in the returned string.

| Horse | Runs: HrCountBy(,'HrDistance()') | Wins: HrCountBy('HrFp()=1','HrDistance()') | Wins: HrCountBy(,'HrDistance()',,'For HrFP()=1') |
|---|---|---|---|
| Wise Lover | 1000(1) 1200(3) 1400(6) 1600(1) | 1000(0) 1200(1) 1400(1) 1600(0) | 1200(1) 1400(1) |
| Whiplash | 1000(2) 1200(2) 1400(4) | 1000(0) 1200(0) 1400(2) | 1400(2) |
| Lifeline Rapic | 1000(3) 1200(3) 1400(17) 1600(2) | 1000(0) 1200(0) 1400(2) 1600(0) | 1400(2) |
| Progressing | 1000(7) 1200(8) 1400(6) 1650(12) | 1000(0) 1200(0) 1400(0) 1650(0) | |
| Master Majo | 1400(1) | 1400(0) | |
| Lucky Dansk | 1000(2) 1200(6) 1400(10) 1650(1) | 1000(0) 1200(1) 1400(2) 1650(0) | 1200(1) 1400(2) |
| Siameric Te | 1200(6) 1400(1) 1650(2) | 1200(3) 1400(0) 1650(0) | 1200(3) |
| Lucky Dragor | 1000(1) 1200(1) 1400(16) 1600(1) 1650(2) | 1000(0) 1200(0) 1400(2) 1600(0) 1650(0) | 1400(2) |
| Honour Supr | 1000(4) 1200(24) 1400(6) 1600(1) 1650(3) | 1000(0) 1200(5) 1400(0) 1600(0) 1650(0) | 1200(5) |
| Super Snippe | 1400(1) | 1400(0) | |
| Spokesman | 1000(1) 1200(12) 1400(11) 1600(2) 1650(6) | 1000(0) 1200(0) 1400(1) 1600(0) 1650(2) | 1400(1) 1650(2) |
| Classabove | 1400(3) | 1400(0) | |
| Multi-win Spi | 1200(3) | 1200(0) | |
| Global Way | 1400(2) | 1400(0) | |

### Enhanced Syntax 1: Multiple Statistical Items in Argument 'lCountIf'

```
'..<<lCountIf1>>..<<lCountIf2>>...'
```

You may put more than one statistical item in the 'lCountIf' argument. Each is quoted by a pair of '<<' and '>>'. You may insert any number of characters outside the quotes.

### Example

```
HrCountBy('<<HrFp( )=1>> wins in <<HrFp( )>> starts' ,'HrDistance( )')
```

The 'lCountIf' argument of this example contains 2 statistical items:

1. <<HrFp( )=1>> counts number of wins (i.e., HrFp( )=1) in each distance group
2. <<HrFp( )>> counts number of valid starts, i.e., HrFp( ) is not null.

If a horse has 5 1000m starts, 9 1200m starts and 3 1400m starts and among these starts it won in 1 1000m start, 2 1200m starts, the function will return the following string:

"1000(1 wins in 5 starts) 1200(2 wins in 9 starts) 1400(0 wins in 3 starts)"

### Enhanced Syntax 2: A Summary Header Section in Argument 'lCountIf'

The 'lCountIf' argument can be divided into 2 sections using the separator '||'. The

© Welkin Systems Limited

section before the '||' is a summary section. The statistical items in this section are applied to all starts which satisfy the condition specified by the 'fCondition' argument.

```
'..<<lOverallCountIf1>>..<<lOverallCountIf2>>...||..<<lDetailCountIf1>>..<<lDetailCountIf2>>..
```

| Section | Position | Function |
|---|---|---|
| Summary Section<br>`..<<lOverallCountIf1>>..<<lOverallCountIf2>>...` | Left of \|\| | Statistical items are for all records rather than individual breakdown groups |
| Detail Section<br>`..<<lDetailCountIf1>>..<<lDetailCountIf2>>...` | Right of \|\| | Statistical items are for individual breakdown groups |

You may put a single character "~" in the summary section. The functions will copy the statistics items and text around the items into the summary section. The word "All" will be printed at the position for breakdown values.

**Examples**

```
HrCountBy('Totally <<HrFp( )=1>> wins in <<HrFp( )>> races: ||<<HrFp( )=1>> wins in <<HrFp( )>>
```

If a horse has 5 1000m starts, 9 1200m starts and 3 1400m starts and among these starts it won in 1 1000m start and 2 1200m starts, the function will return the following string:

"Totally 3 wins in 17 races: 1000(1 wins in 5 starts) 1200(2 wins in 9 starts) 1400(0 wins in 3 starts)"

```
HrCountBy('~||<<HrFp( )=1>> wins in <<HrFp( )>> starts' ,'HrDistance( )')
```

Provided that you are applying the function to the same horse and races as those in the prior example, you will get the following return string:

```
"All(3 wins in 17 starts) 1000(1 wins in 5 starts) 1200(2 wins in 9 starts) 1400(0 wins in 3 s
```

**Enhanced Syntax 3: Statistical Function Modifiers**

You may precede a statistical item with one of the following modifiers:

| Modifier | Function |
|---|---|
| CNT: | Counts records in which the value of the expression after "CNT:" is either true or not Null. |
| CNTF: | Similar to modifier "CNT:" except that if the statistics result of item is 0, the result of the related breakdown group will be removed. |
| CNT%: | Percentage of records counted (the expression after CNT% returns either true or not Null) in the breakdown group. |
| CNTF%: | Similar to CNT% except that if the statistics result of item is 0, the result of the related breakdown group will be removed. |
| SUM: | Calculate the sum the values of the expression following the modifier over each breakdown group. Null value will be skipped. |
| AVE: | Calculate the average the values of the expression following the modifier over each breakdown group. Null value will be skipped. |

© Welkin Systems Limited

| | |
|---|---|
| MAX: | Find the maximum value of the expression following the modifier over each breakdown group. Null value will be skipped. |
| MIN: | Find the minimum value of the expression following the modifier over each breakdown group. Null value will be skipped. |
| ANY: | Return an expression value of an arbitrary record in each breakdown group |

**Example**

```
HrCountBy('Wins:<<Cnt:HrFp( )=1>>,Best:<<Min:HrFp( )>>','HrDraw( )')
```

Or

```
HrCountBy('Wins:<<HrFp( )=1>>,Best:<<Min:HrFp( )>>','HrDraw( )')
```

The formulae above return the following 2 statictics of each draw position group:

1. Number of wins <<Cnt:HrFp( )=1>>. Since "Cnt" is the default behaviour of HrCountBy( ), the modifier "Cnt:" is optional. Therefore, results of the above two functions are identical.
2. Best finishing position <<Min:HrFp( )>>

Below is an example of the return string of the above functions:

```
"1(Wins:0,Best:3) 2(Wins:1,Best:1) 6(Wins:0,Best:12) 10(Wins:2,Best:1) "
```

```
HrCountBy('Wins:<<Cntf:HrFp( )=1>>,Best:<<Min:HrFp( )>>','HrDraw( )')
```

The statistics done in this example is basically identical to what is done in the functions of the previous example. The difference is if the count of the first statistical item <<Cntf:HrFp( )=1>> is 0, the breakdown group will be removed. If we apply the above function to exactly the same horse and race as prior example, the string returned will be as follows (please compare with prior example):

```
"2(Wins:1,Best:1) 10(Wins:2,Best:1) "
```

**Enhanced Syntax 4: Customize Format of Group Caption in Argument 'eBreakdownExpr'**

By default, statistical result of each breakdown group is enclosed by a pair of brackets, which is preceded by the value of the breakdown expression, i.e.: xxxx(...), where "xxxx" is the value returned by eBreakdownExpr and "...." is the result returned by the "lCountIf" argument. Following the rules to replace this default format:

1. Enclose the breakdown expression with the quotes "<<" and ">>", which will act as a placeholder for the breakdown value.
2. Enter the text that you want to replace the default caption outside the quotes.
3. Once you have use the "<<" and ">>" quotes, the default brackets will no longer appear.

**Example**

```
HrCountBy('Wins:<<HrFp( )=1>>,Best:<<Min:HrFp( )>>  ','Draw#<<HrDraw( )>>:')
```

The return value of the function will be similar to the following:

```
"Draw#1:Wins:0,Best:3  Draw#2:Wins:1,Best:1  Draw#6:Wins:0,Best:12  Draw#10:Wins:2,Best:1  "
```

© Welkin Systems Limited

Please note that there are 2 spaces at the end of the first argument so that groups will not be too close to each other.

### Enhanced Syntax 5: Embedding an Sorting Expression in Argument 'eBreakdownExpr'

You may define an expression for sorting breakdown groups and put it at the very beginning of the 'eBreakdownExpr' argument. The expression must be quoted by the brackets "{ }" and begins with "ASC:" or "DESC:" (for ascending and descending sorts). You may embed statistics items inside the sort expression. These statistics items must be enclosed by "<< >>". The format of the items is similar to a normal statistics item as described in the prior sections.

### Examples

```
HrCountBy('HrFp( )=1','{Desc:<<Cnt:HrFp( )=1>>}HrDistance( )')
```

Returns "1400(6) 1200(4) 1000(1) 1600(0)"

```
HrCountBy('HrFp( )=1','HrDistance( )')
```

Returns "1000(1) 1200(4) 1400(6) 1600(0)"

The functions above carry out the same statistics. However, an expression for sorting is specified in the first function, which makes the groups being arranged in descending order of the win counts.

### Enhanced Syntax 6: New String to Replace the Default Separator between 2 Statistical Result Items

You may define a new string to replace the default separator (space) between 2 statistical result items. Put a "||" to the right of the breakdown expression followed by the string you want to replace the default separator. The following examples shows the syntax. You may include any number of expressions in your new separator. All expressions must be quoted with "<<" and ">>".

### Examples

```
HrCountBy('HrFp( )=1','HrDistance( )|| / ')
```

The function returns "1000(1) / 1200(4) / 1400(6) / 1600(0)". The new separator " / " is defined at the right of "||".

```
HrCountBy('HrFp( )=1','HrDistance( )||<<Chr(13)>>')
```

New line character Chr(13) is defined as the new separator. It separates statistical items into new lines. Please note that Chr(13) is an expression and must be enclosed with "<<" and ">>". The function returns the following 4 lines.
1000(1)
1200(4)
1400(6)
1600(0)

© Welkin Systems Limited

**See also**

HrCount( ), JrCount( ), SrCount( ) | HrCountNZ( ), JrCountNZ( ), SrCountNZ( ) | HrAverageBy( ), JrAverageBy( ), SrAverageBy( ) | HrSumBy( ), JrSumBy( ), SrSumBy( )

## 16.8.6 HrCountNZ( ), JrCountNZ( ), SrCountNZ( )

This set of functions is similar to the HrCount( ) function set. However, these functions will skip items with values Null and empty values (0, empty string or empty date etc). (NZ stands for No Zero)

**Syntax**

```
HrCountNZ('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
JrCountNZ('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
SrCountNZ('lCountIf',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'lCountIf' | • Datum the count of which is to be found.<br>• lCountIf can be of any data type. If lCountIf is of logical type, all true (.T.) values will be counted. For any other types, record will be counted only if lCountIf returns a non-null value. | 'HrFP( )=1' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be | |

© Welkin Systems Limited

| | | |
|---|---|---|
| | found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Numeric

**Examples**

```
HrCountNz('HrRating( )>80',6,'For HrDistance( )=1400')
HrCount('HrRating( )>80',6,'For HrDistance( )=1400')
```

The above 2 examples will return exactly the same result.

```
HrCountNz('HrRating( )',6,'For HrDistance( )=1400')
```

The above example returns the number of 1400m starts, in which HrRating( ) returns a non-null and non-zero value, in the past record of the horse. If total number of races is greater than 6, the function will return 6 instead of the actual number. Please note that starts will NOT be counted if HrRating( ) returns 0.

**See also**

HrCount( ), JrCount( ), SrCount( )

## 16.8.7 HrMaximum( ), JrMaximum( ), SrMaxmum( )

The functions return the maximum of a racing datum in the past records.

**Syntax**

```
HrMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
JrMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
SrMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the maximum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| | 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| eDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Data type identical to the expression passed into the first argument.

**Examples**

```
HrMaximum('HrWeight( )',3)
```

The above example returns the maximum loading of the horse in last 3 starts.

```
HrMaximum('HrWeight( )',4,'For HrDistance( )=1400')
```

The above example returns the maximum loading of the horse in last 4 1400m races.

```
HrMaximum('HrWeight( )',,'For HrDistance( )=1400 and HrAveRating( )>100',101)
```

The above example returns the maximum loading of the horse in all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

If no races satisfying the above conditions can be found 101 will be returned.

© Welkin Systems Limited

### 16.8.8 HrMaxStartID( ), JrMaxStartID( ), SrMaxStartID( )

The functions return ID of the start at which the maximum of a racing datum is found.

#### Syntax

```
HrMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the maximum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

#### Returns

Character (Width=8)

© Welkin Systems Limited

### Examples

```
HrMaxStartID('HrWeight( )',3)
```

The above example returns the ID of the start, among the last 3, at which the loading is a maximum.

```
HrMaxStartID('HrWeight( )',4,'For HrDistance( )=1400')
```

The above example returns the ID of the start, among the last 4 1400m races, at which the loading of the horse is a maximum.

```
HrMaxStartID('HrWeight( )',,'For HrDistance( )=1400 and HrAveRating( )>100',101)
```

The above example returns ID of the start, among the races satisfying the following conditions, at which the loading of the horse is a maximum:

1. 1400m race
2. Average rating of the race is greater than 100

If no races satisfying the above conditions can be found 101 will be returned.

## 16.8.9 HrMinimum( ), JrMinimum( ), SrMinimum( )

The functions return the minimum of a racing datum in the past records.

### Syntax

```
HrMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
JrMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
SrMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the minimum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | • A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | |
| eDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Data type identical to the expression passed into the first argument.

**Examples**

```
HrMinimum('HrWeight( )',3)
```

The above example returns the minimum loading of the horse in last 3 starts.

```
HrMinimum('HrWeight( )',4,'For HrDistance( )=1400')
```

The above example returns the minimum loading of the horse in last 4 1400m races.

```
HrMinimum('HrWeight( )',,'For HrDistance( )=1400 and HrAveRating( )>100',101)
```

The above example returns the minimum loading of the horse in all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

If no races satisfying the above conditions can be found 101 will be returned.

## 16.8.10 HrMinStartID( ), JrMinStartID( ), SrMinStartID( )

The functions return ID of the start at which the minimum of a racing datum is found.

**Syntax**

```
HrMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the minimum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| | from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character (Width=8)

**Examples**

```
HrMinStartID('HrWeight( )',3)
```

The above example returns the ID of the start, among the last 3, at which the loading is a minimum.

```
HrMinStartID('HrWeight( )',4,'For HrDistance( )=1400')
```

The above example returns the ID of the start, among the last 4 1400m races, at which the loading of the horse is a minimum.

```
HrMinStartID('HrWeight( )',,'For HrDistance( )=1400 and HrAveRating( )>100',101)
```

The above example returns ID of the start, among the races satisfying the following conditions, at which the loading of the horse is a minimum:

© Welkin Systems Limited

1. 1400m race
2. Average rating of the race is greater than 100

If no races satisfying the above conditions can be found 101 will be returned.

## 16.8.11 HrStatList( ), JrStatList( ), SrStatList( )

The functions return sum, maximum, minimum, and count of a racing datum in the past records. The 2 figures are returned in a single string separated with a ','. For example, a returned value "1324,16,8,11" means that 1324 is got by summing up 11 races and the maximum and minimum value in these 11 races are 16 and 8 respectively.

Using these functions, you can get sum, maximum, minimum, count, and average (sum/count) by only scanning the database once. It is much faster than applying HrSum( ), HrMaximum, HrMinimum, HrCount( ), and HrAverage( ) in

separate columns.

Use the ListVal( ) function to retrieve a specific item in the list.

### Syntax

```
HrStatList('nValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the statistics of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |

© Welkin Systems Limited

| | | |
|---|---|---|
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character

## 16.8.12 HrSum( ), JrSum( ), SrSum( )

The functions return the sum of a racing datum over a set of the past records of a horse, a jockey or a stable.

**Syntax**

```
HrSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
JrSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
SrSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the sum of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | evaluates to false.<br>• Both For and While clauses are optional. | |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Numeric

**Examples**

```
HrSum('HrWinTickets( )',3)
```

The above example returns the sum of win investment of the last 3 races of the horse.

```
HrSum('HrWinTickets( )',4,'For HrDistance( )=1400')
```

The above example returns the sum of win investment of the last 4 1400m races of the horse.

```
HrSum('HrWinTickets( )',,'For HrDistance( )=1400 and HrAveRating( )>100',95003)
```

The above example returns the sum of win investment of all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

## 16.8.13 HrSumBy( ), JrSumBy( ), SrSumBy( )

The functions sum a value over past records, which satisfy a specific condition, of the horse, jockey and stable respectively. The results will be broken down by the value of an expression and returned as a character string.

**Syntax**

```
HrSumBy('nValueExpr','eBreakdownExpr',nRraces,'fCondition',cDefaultValue,cIncluding)
JrSumBy('nValueExpr','eBreakdownExpr',nRraces,'fCondition',cDefaultValue,cIncluding)
SrSumBy('nValueExpr','eBreakdownExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

**Enhanced Syntax**

By using the enhanced syntax you may:

1. carry out multiple statistics in a single pass of past record scanning
2. customize the format of return string
3. define the order of breakdown groups

© Welkin Systems Limited

4. define a summary session in the return string

The enhanced syntax of these functions is similar to that for HrCountBy( ), JrCountBy ( ), SrCountBy( ). Please see its help page for details.

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the sum of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| 'eBreakdownExpr' | An expression by which results will be broken down. Omitting this argument tells the function do the statistics over all legitimate records. | 'HrDistance( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character (The length of the returned character string depends on the breakdown

© Welkin Systems Limited

expression. It is recommended to set the column data type to Multiline Text, so that the number of characters returned can be of any length.)

**Examples**

```
HrSumBy('HrWinTickets( )','HrDistance( )')
```

The above function sums the win tickets of a horse in all races. The result will then be broken down by distance.

```
HrSumBy('HrWinTickets( )','HrDistance( )',, 'HrFp( )=1')
```

The above function sums the win tickets of a horse in all winning races. The result will then be broken down by distance.

**See also**

HrCountBy( ), JrCountBy( ), SrCountBy( ) | HrAverageBy( ), JrAverageBy( ), SrAverageBy( ) | HrSum( ), JrSum( ), SrSum( )

## 16.8.14 HrSumCount( ), JrSumCount( ), SrSumCount( )

The functions return both sum and count of a racing datum in the past records. The 2 figures are returned in a single string separated with a ','. For example, a returned value "1324,11" means that 1324 is got by summing up 11 races.

Using these functions, you can get sum, count, and average (sum/count) by only scanning the database once. It is much faster than applying HrSum( ), HrCount( ), and HrAverage( ) in separate columns.

Use the ListVal( ) function to retrieve a specific item in the list.

**Syntax**

```
HrSumCount('nValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the sum and count of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character

**Examples**

```
HrSumCount('HrWinTicket( )',3)
```

The above example returns the sum and count of win investment of the last 3 races of the horse.

If the above function is put into a column 'WinTK', the following functions will returns the sum, count, and average from the result respectively:

Sum: `ListVal(Column('WinTk'),1)`
Count: `ListVal(Column('WinTk'),2)`
Average: `Column('Sum')/Column('Count')`

```
HrSum('HrWinTicket( )',4,'For HrDistance( )=1400')
```

The above example returns the sum and count of win investment of the last 4 1400m races of the horse.

```
HrSum('HrWinTicket( )',,'For HrDistance( )=1400 and HrAveRating( )>100',95003)
```

The above example returns the sum of win investment of all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

If there is no past race satisfying the condition, the default value 95003 will be returned.

© Welkin Systems Limited

## 16.8.15 HrXAverage( ), JrXAverage( ), SrXAverage( )

The above functions are similar to functions HrAverage( ), JrAverage( ), and SrAverage ( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although the functions accept 5 arguments, the 4th argument of HrXAverage( ), JrXAverage( ), SrXAverage( ) will always be ignored.

### Syntax

```
HrXAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
JrXAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
SrXAverage('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the average of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You | 'R' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | may use this argument to change the scope. See the following page for details: The cIncluding Argument. | |

**Returns**

Numeric

**Examples**

```
HrXAverage('HrWeight( )',3)
```

The above example returns the average loading of the past 3 races of the horse.

```
HrAverage('HrSpeed( )',4,'For HrDistance( )=1400')
```

The above example returns the average speed of the last 4 1400m races of the horse.

```
HrXAverage('HrSpeed( )',,'For HrDistance( )=1400 and HrAveRating( )>100')
```

The above example returns the average speed of all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

### 16.8.16 HrXMaximum( ), JrXMaximum( ), SrXMaximum( )

The above functions are similar to functions HrMaximum( ), JrMaximum( ), and SrMaximum( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although the functions accept 5 arguments, the 4th argument of HrXMaximum( ), JrXMaximum( ), SrXMaximum( ) will always be ignored.

**Syntax**

```
HrXMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
JrXMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
SrXMaximum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the maximum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| | the HrCount( ) function to check how many races have been counted. | |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| eDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Data type of eValueExpr

## 16.8.17 HrXMaxStartID( ), JrXMaxStartID( ), SrXMaxStartID( )

The above functions are similar to functions HrMaxStartID( ), JrMaxStartID( ), and SrMaxStartID( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although the functions accept 5 arguments, the 4th argument of HrXMaxStartID( ), JrXMaxStartID( ), SrXMaxStartID( ) will always be ignored.

**Syntax**

```
HrXMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrXMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrXMaxStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the maximum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be | 6 |

© Welkin Systems Limited

| | | |
|---|---|---|
| | averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Character

## 16.8.18 HrXMinimum( ), JrXMinimum( ), SrXMinimum( )

The above functions are similar to functions HrMinimum( ), JrMinimum( ), and SrMinimum( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although the functions accept 5 arguments, the 4th argument of HrXMinimum( ), JrXMinimum( ), SrXMinimum( ) will always be ignored.

**Syntax**

```
HrXMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
JrXMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
SrXMinimum('eValueExpr',nRaces,'fCondition',eDefaultValue,cIncluding)
```

© Welkin Systems Limited

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the minimum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| eDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

**Returns**

Data type of eValueExpr

## 16.8.19 HrXMinStartID( ), JrXMinStartID( ), SrXMinStartID( )

The above functions are similar to functions HrMinStartID( ), JrMinStartID( ), and SrMinStartID( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although the functions accept 5 arguments, the 4th argument of HrXMinStartID( ), JrXMinStartID( ), SrXMinStartID( ) will always be ignored.

© Welkin Systems Limited

### Syntax

```
HrXMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
JrXMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
SrXMinStartID('eValueExpr',nRaces,'fCondition',cDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the minimum of which is to be found.<br>• Data type of expression: Any | 'HrDvdWin( )' |
| 'eBreakdownExpr' | An expression by which results will be broken down. Omitting this argument tells the function do the statistics over all legitimate records. | 'HrDistance( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| cDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You may use this argument to change the scope. See the following page for details: The cIncluding Argument. | 'R' |

### Returns

© Welkin Systems Limited

Character

## 16.8.20 HrXSum( ), JrXSum( ), SrXSum( )

These functions is similar to the functions HrSum( ), JrSum( ), and SrSum( ) except:

1. If the number of races meeting 'fCondition' is smaller than nRaces, the function will return .NULL.
2. Although these functions accept 5 arguments, the 4 th argument will always be ignored.

### Syntax

```
HrXSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
JrXSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
SrXSum('nValueExpr',nRaces,'fCondition',nDefaultValue,cIncluding)
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the sum of which is to be found.<br>• Data type of expression: Numeric | 'HrDvdWin( )' |
| nRaces | • Number of latest races, counting back from the underlying race, to be averaged. The races must be meeting the condition specified by 'fCondition'.<br>• If nRaces is omitted, all races meeting 'fCondition' will be included.<br>• If total number of races meeting 'fCondition' is smaller than nRaces, the function will perform calculations on only the available races. You may use the HrCount( ) function to check how many races have been counted. | 6 |
| 'fCondition' | • If argument 'fCondition' is passed, the races (nRaces) will only be selected from those meeting 'fCondition'. The 'fCondition' argument is composed of a For clause and a While clause.<br>• A For clause causes the function to act on each record that meets the specified logical condition.<br>• A While clause causes the function to stop action once the logical expression evaluates to false.<br>• Both For and While clauses are optional. | 'For HrFP( )=1 While HrDate( )>Date(1999,9,1)' |
| nDefaultValue | Default value returned when no race satisfying the specified condition can be found. The data type of this value must be identical to that of normal return value. | |
| cIncluding | When applying the function to past records, by default, races in the underlying meeting are not included. You | 'R' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | may use this argument to change the scope. See the following page for details: The cIncluding Argument. | |

**Returns**

Numeric

**Examples**

```
HrXSum('HrWinTicket( )',3)
```

The above example returns the sum of win investment of the last 3 races of the horse.

```
HrXSum('HrWinTicket( )',4,'For HrDistance( )=1400')
```

The above example returns the sum of win investment of the last 4 1400m races of the horse.

```
HrXSum('HrWinTicket( )',,'For HrDistance( )=1400 and HrAveRating( )>100',95)
```

The above example returns the sum of win investment of all races meeting the following conditions:

1. 1400m race
2. Average rating of the race is greater than 100

## 16.8.21 StdTime( )

This function returns the averages of running time (standard time) of all races particular combinations of distance, course, track, and class.

**Syntax**

```
StdTime(nBackSeason,nDist,cClass,cCourse,cGoingID)
```

| Argument | Description | Example |
|---|---|---|
| nBackSeason | Target season. Default to season of underlying race. | 1 (for last season) |
| nDist | Distance of target races. Default to distance of underlying race. | 1400 (1400m races) |
| cClass | Class of target races. Default to class of underlying race. | '2' (for 2nd class) |
| cCourse | Course of target races. Default to course of underlying race. | 'H' (for Happy Valley) |
| cGoingID | Going ID of target races. Default to going of underlying race. | '1' (for firm going) |

**Returns**

Character (Width=6)

Example of returned value: '1.36.3'

© Welkin Systems Limited

**Example**

The following function returns the standard time distance, class, course and going similar to those of the underlying race in the underlying season.

```
StdTime( )
```

The following function returns the standard time distance, class, course and going similar to those of the underlying race in last season.

```
StdTime(1)
```

Assume you are opening a first class race. The following function returns the standard time for the class 2 in the underlying season with distance, course and going identical to the current race:

```
StdTime(1, ,'2')
```

### 16.8.22 StdTimew( )

This function is identical to its original version, except that they return 2 digits after the decimal point. Therefore the width of the return value is one character wider than that of the original function. Click on the link below to see the syntax of the function.

StdTime( )

**Returns**

Character (Width=7)

## 16.9 Statistics in a Race

This group of functions returns statistical figures on races.

**>>> Subtopics of this section <<<**

- RunnersAvg( )
- RunnersCnt( )
- RunnersFind( )
- RunnersList( )
- RunnersMax( )
- RunnersMin( )
- RunnersSum( )

### 16.9.1 RunnersAvg( )

The RunnersAvg( ) function returns the average of a datum about a runner in a particular race.

**Syntax**

```
RunnersAvg('nValueExpr',cRaceID,'fCondition')
```

© Welkin Systems Limited

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the average of which is to be found.<br>• Data type of expression: Numeric | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

**Returns**

The function returns a numeric value.

**Examples**

```
RunnersAvg('Bodyweight( )')
```

Returns the average Body Weight of the underlying race.

```
RunnerAvg('Bodyweight( )','17222')
```

Returns the average Body Weight of the race of ID '17222', i.e., the 222nd race of season 97-98.

```
RunnerAvg('Bodyweight( )',,'HrFP( )<=3')
```

Returns the average Body Weight amount the 3 fastest horses in the underlying race.

```
RunnerAvg('Bodyweight( )','17222','HrFP( )<=3')
```

Returns the average Body Weight amount the 3 fastest horses of the race of ID '17222', i.e., the 222 nd race of season 97-98.

## 16.9.2 RunnersCnt( )

The RunnersCnt( ) function returns the number of runners that meets a particular condition in a race.

**Syntax**

```
RunnersCnt('eValueExpr',cRaceID,'fCondition')
```

| Argument | Description | Example |
|---|---|---|
| 'lCountIf' | • Datum the count of which is to be found.<br>• lCountIf can be of any data type. If lCountIf is of logical type, all true (.T.) values will be counted. For any other types, record will be counted only if lCountIf returns a non-null value. | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions | '17222' |

© Welkin Systems Limited

| | | |
|---|---|---|
| | HrRaceID( ), JrRaceID( ) and SrRaceID( ). | |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

**Returns**

The function returns a numeric value.

**Examples**

```
RunnersCnt( )
```

Returns the number of runners in the underlying race.

```
RunnersCnt('17222')
```

Returns number of runners in the race of ID '17222', i.e., the 222nd race of season 97-98.

```
RunnersCnt(,'HrRating( )>95')
```

Returns number of runners of rating greater than 95 in the underlying race.

```
RunnersCnt('17222','HrRating( )>95')
```

Returns number of runners of rating greater than 95 in the race of ID '17222'

i.e., the 222nd race of season 97-98.

### 16.9.3 RunnersFind( )

The functions return the value of an MCL function expression at a start that satisfies a particular condition. Searching will start from the winning horse.

**Syntax**

```
RunnerFind('nValueExpr',cRaceID,'fCondition',nNth)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the location of which is to be found.<br>• Data type of expression: Any | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

**Returns**

Data type identical to that of the first argument.

**Example**

```
RunnersFind('HrHorse( )',17222,'For HrBleedDate( )<>{}')
```

© Welkin Systems Limited

Returns the name of the horse, if any, which bleed in the race of ID 17222 (season 97-98 the 222nd race)

### 16.9.4 RunnersList( )

The function returns a string composed of a list of items got by concatenating the values of an MCL expression evaluated against the horses in a race.

#### Syntax

```
RunnerList('nValueExpr',cRaceID,'fCondition',cSeparator)
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the value list of which is to be found.<br>• Data type of expression: Any | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

#### Returns

Character

#### Example

```
RunnersList('HrHorse( )',HrRaceID(1),'HrFP( )<=2')
```

Returns the names of the winner and the runner-up of a race.

```
RunnersList('HrRating( )',17222,'For HrFP( )<=3','-')
```

Returns a string showing the ratings of the 3 fastest horses in the race of ID 17222. Use "-" as separator. The returned string will be similar to "71-80-77".

### 16.9.5 RunnersMax( )

The RunnersMax( ) function returns the maximum value of a datum about a runner in a particular race.

#### Syntax

```
RunnersMax('nValueExpr',cRaceID,'fCondition')
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the maximum of which is to be found.<br>• Data type of expression: Any | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |

© Welkin Systems Limited

| | | |
|---|---|---|
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

**Returns**

The type of the value returned by the function is identical to that of the first argument of the function.

**Examples**

```
RunnersMax('Bodyweight( )')
```

Returns the maximum Body Weight of the underlying race.

```
RunnersMax('Bodyweight( )','17222')
```

Returns the maximum Body Weight of the race of ID '17222', i.e., the 222nd race of season 97-98.

```
RunnersMax('Bodyweight( )',,'HrFP( )<=3')
```

Returns the maximum Body Weight amount the 3 fastest horses in the underlying race.

```
RunnersMax('Bodyweight( )','17222','HrFP( )<=3')
```

Returns the maximum Body Weight amount the 3 fastest horses of the race of ID '17222', i.e., the 222nd race of season 97-98.

### 16.9.6 RunnersMin( )

The RunnersMin( ) function returns the minimum value of a datum about a runner in a particular race.

**Syntax**

```
RunnersMin('nValueExpr',cRaceID,'fCondition')
```

| Argument | Description | Example |
|---|---|---|
| 'eValueExpr' | • Datum the minimum of which is to be found.<br>• Data type of expression: Any | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

**Returns**

The type of the value returned by the function is identical to that of the first argument of the function.

**Examples**

```
RunnersMin('Bodyweight( )')
```

© Welkin Systems Limited

Returns the minimum Body Weight of the underlying race.

```
RunnersMin('Bodyweight( )','17222')
```

Returns the minimum Body Weight of the race of ID '17222', i.e., the 222nd race of season 97-98.

```
RunnersMin('Bodyweight( )',,'HrFP( )<=3')
```

Returns the minimum Body Weight amount the 3 fastest horses in the underlying race.

```
RunnersMin('Bodyweight( )','17222','HrFP( )<=3')
```

Returns the minimum Body Weight amount the 3 fastest horses of the race of ID '17222', i.e., the 222nd race of season 97-98.

## 16.9.7 RunnersSum( )

The RunnersSum( ) function returns the sum of a datum of all the runners in a particular race.

### Syntax

```
RunnersSum('nValueExpr',cRaceID,'fCondition')
```

| Argument | Description | Example |
|---|---|---|
| 'nValueExpr' | • Datum the sum of which is to be found.<br>• Data type of expression: Numeric | 'HrBodyweight( )' |
| cRaceID | Race ID (default to underlying race). Normally, this is a string returned by the functions HrRaceID( ), JrRaceID( ) and SrRaceID( ). | '17222' |
| 'fCondition' | Logical expression to filter out a particular set of runners in the race. | 'HrFP( )<=3' |

### Returns

The function returns a numeric value.

### Examples

```
RunnersSum('Bodyweight( )')
```

Returns the sum of Body Weight of all the runners of the underlying race.

```
RunnersSum('Bodyweight( )','17222')
```

Returns the sum of Body Weight of all the runners of the race of ID '17222', i.e., the 222nd race of season 97-98.

```
RunnersSum('Bodyweight( )',,'HrFP( )<=3')
```

Returns the sum of Body Weight of the 3 fastest horses in the underlying race.

```
RunnersSum('Bodyweight( )','17222','HrFP( )<=3')
```

© Welkin Systems Limited

Returns the sum of Body Weight of the 3 fastest horses of the race of ID '17222', i.e., the 222nd race of season 97-98.

## 16.10 Arithmetic Functions

This group of functions performs arithmetic operations.

**>>> Subtopics of this section <<<**

- Average( )
- AverageNZ( )
- Int( )
- Maximum( )
- Minimum( )
- Over( )
- Sum( )

### 16.10.1 Average( )

The Average( ) function returns the average of a set of values. Null values inside the set will be ignored. You may pass up to 27 arguments into the function.

**Syntax**

```
Average(x1,x2,x3,...)
```

x1, x2, x3,... must be of numeric type.

**Returns**

Numeric

**Examples**

```
Average(Column('This Rating'),Column('Last Rating'))
```

The average of the columns This Rating and Last Rating will be returned.

```
Average(12, .NULL., 24)
```

18 will be returned by the above function. .NULL. is ignored.

```
Average(12, 0, 24)
```

12 will be returned by the above function.

### 16.10.2 AverageNZ( )

The AverageNZ( ) function returns the average of a set of values. Zero values will not be counted (You may pronouns NZ as No Zero). You may pass up to 27 arguments into the function.

© Welkin Systems Limited

**Syntax**

```
AverageNZ(x1,x2,x3,...)
```

x1, x2, x3,... must be numbers.

**Returns**

Numeric

**Examples**

```
AverageNZ(Column('This Rating'),Column('Last Rating'))
```

The average of the columns This Rating and Last Rating will be returned.

```
AverageNZ(2, 0, .Null., 4)
```

3 will be returned by the above function (average of 2 and 4). Both 0 and .NULL. are ignored.

### 16.10.3 Int( )

A Visual FoxPro built-in function, which returns the integer part of a number.

**Syntax**

```
Int(nExpr)
```

**Returns**

Numeric

**Example**

```
Int(3.75)
```

Returns integer 3

### 16.10.4 Maximum( )

The Maximum( ) function returns the maximum of a set of values. Null values inside the set will be ignored.

**Syntax**

```
Maximum(x1,x2,x3,...)
```

x1, x2, x3,... must be of the same data type. You may pass up to 27 arguments into the function.

**Returns**

Value returned by the function will be of the same type and width of the referencing column.

© Welkin Systems Limited

#### Examples

```
Maximum(Column('This Rating'),Column('Last Rating'))
```

The maximum of the columns This Rating and Last Rating will be returned.

```
Maximum(123, .NULL., 234)
```

234 will be returned by the above function. .NULL. is ignored.

```
Maximum('A+', 'B')
```

'B' will be returned by the above function. A+ is smaller than B in alphabetical order.

### 16.10.5 Minimum( )

The Minimum( ) function returns the minimum of a set of values. Null values inside the set will be ignored. You may pass up to 27 arguments into the function.

#### Syntax

```
Minimum(x1,x2,x3,...)
```

x1, x2, x3,... must be of the same data type.

#### Returns

Value returned by the function will be of the same type and width of the referencing column.

#### Examples

```
Minimum(Column('This Rating'),Column('Last Rating'))
```

The minimum of the columns This Rating and Last Rating will be returned.

```
Minimum(123, .NULL., 234)
```

123 will be returned by the above function. .NULL. is ignored.

```
Minimum('A+', 'B')
```

'A+' will be returned by the above function. A+ is smaller than B in alphabetical order.

### 16.10.6 Over( )

The function returns the result of an divide operation of 2 values. Null values will be returned if the denominator is zero.

#### Syntax

```
Over(x1,x2)
```

#### Returns

Returns the result of an divide operation of 2 values. Null values will be returned if the denominator is zero.

© Welkin Systems Limited

#### Examples

```
Over(3,4)
```

Returns 0.75

```
Over(3,0)
```

Returns NULL.

### 16.10.7 Sum( )

The Sum( ) function returns the sum of a set of values. Null values inside the set will be ignored. You may pass up to 27 arguments into the function.

#### Syntax

```
Sum(x1,x2,x3,...)
```

x1, x2, x3,... must be numbers.

#### Returns

Numeric

#### Examples

```
Sum(Column('This Rating'),Column('Last Rating'))
```

The sum of the columns This Rating and Last Rating will be returned.

```
Sum(12, .NULL., 24)
```

36 will be returned by the above function. .NULL. is ignored.

```
Sum(12, 0, 24)
```

36 will be returned by the above function.

## 16.11 String Functions

This group of functions is used for manipulating character data.

#### >>> Subtopics of this section <<<

- At( )
- Atc( )
- Left( )
- Like( )
- LTrim( )
- Occurs( )
- Padr( )
- Replicate( )
- Right( )
- StrExtract( )

© Welkin Systems Limited

- SubStr( )
- Trim( )

### 16.11.1 At( )

A Visual FoxPro built-in function which returns a number of telling the position of a characters expression in another character expression.

#### Syntax

```
At(cSearchExpression, cExpressionSearched, nOccurrence)
```

#### Argument

cSearchExpression

Specifies the character expression to search for in cExpressionSearched.

cExpressionSearched

Specifies the character expression to search for.

#### Returns

Numeric

### 16.11.2 Atc( )

A Visual FoxPro built-in function which returns a number of telling the position of a characters expression in another character expression. This function is exactly the same as the At( ), except that the searching will be case insensitive.

#### Syntax

```
Atc(cSearchExpression, cExpressionSearched, nOccurrence)
```

#### Argument

cSearchExpression

Specifies the character expression to search for in cExpressionSearched.

cExpressionSearched

Specifies the character expression to search for.

#### Returns

Numeric

### 16.11.3 Left( )

A Visual FoxPro built-in function which returns a specified number of characters from a character expression, starting with the leftmost character.

© Welkin Systems Limited

**Syntax**

Left(cExpression, nCharacters)

**Argument**

cExpression

Specifies the character expression from which Left( ) returns characters.

nCharacters

Specifies the number of characters returned from the character expression. If nCharacters is greater than the length of cExpression, all of the character expression is returned. The empty string is returned if nCharacters is negative or 0.

**Returns**

Character

## 16.11.4 Like( )

A Visual FoxPro built-in function which determines if a character expression matches another character expression.

**Syntax**

Like(cTemplate, cExpression)

**Argument**

cTemplate

Specifies a template string that LIKE( ) compares with cExpression. cTemplate can contain the wildcards such as * and ?. The question mark (?) matches any single character in cExpression and the asterisk (*) matches any number of characters. You can mix any number of wildcards in any combination in cTemplate.

cExpression

Specifies the character expression LIKE( ) compares with cTemplate. cExpression must match cTemplate letter for letter in order for LIKE( ) to return true (.T.).

**Returns**

Logical

## 16.11.5 LTrim( )

A Visual FoxPro built-in function which removes all leading spaces or parsing characters from the specified character expression.

**Syntax**

LTrim(cExpression, nFlags, cParseChar, cParseChar2, ...)

© Welkin Systems Limited

**Argument**

cExpression

Specifies an expression of character to remove all trailing spaces.

nFlags

Specifies if trimming is case-sensitive when one or more parse characters (cParseChar, cParseChar2, … are included. Trimming is case-sensitive if nFlags is zero or is omitted. Trimming is case-insensitive if nFlags 1.

cParseChar, cParseChar2, ...

Specifies one or more character strings that are trimmed from the end of cExpression. If cParseChar isn't included, then trailing spaces. The maximum number of strings permitted in cParseChar is 23.

**Returns**

Character

**Examples**

```
LTrim('  ABC/..')
```

Returns 'ABC/..'

```
LTrim('  ../ABC',[ ],[.],[/])
```

Returns 'ABC'

## 16.11.6 Occurs( )

A Visual FoxPro built-in function which returns the number of times a character expression occurs within another character expression.

**Syntax**

```
Occurs(cSearchExpression, cExpressionSearched)
```

**Argument**

cSearchExpression

Specifies a character expression that OCCURS( ) searches for within cExpressionSearched.

cExpressionSearched

Specifies the character expression OCCURS( ) searches for cSearchExpression.

**Return Value**

Numeric

© Welkin Systems Limited

**Example**

```
Occurs('a', 'abraca')
```

Returns 3

```
Occurs('b', 'abraca')
```

Returns 1

```
Occurs('e', 'abraca')
```

Returns 0

## 16.11.7 Padr( )

A Visual FoxPro built-in function which returns a string from an expression, padded with spaces or characters to a specified length on the right sides.

**Syntax**

```
Padr(cExpression,nCharacters,cCharacter)
```

**Argument**

| | |
|---|---|
| cExpression | Expression to be padded. May be string, number or date |
| nCharacters | Length of the returned string |
| cCharacter | Character to use for padding. |

**Returns**

Character

**Example**

```
Padr("Good",5)+"boy"
```

Returns "Good boy"

```
Padr(12,4)+"pc"
```

Returns "12 pc"

## 16.11.8 Replicate( )

A Visual FoxPro built-in function which returns a character string that contains a specified character expression repeated a specified number of times.

**Syntax**

```
Replicate(cExpression, nTimes)
```

© Welkin Systems Limited

**Argument**

cExpression

Specifies the character expression that is replicated.

nTimes

Specifies the number of times the character expression is replicated.

**Returns**

Character

## 16.11.9 Right( )

A Visual FoxPro built-in function which returns the specified number of rightmost characters from a character string.

**Syntax**

```
Right(cExpression,nCharacters)
```

**Argument**

cExpression

Specifies the character expression whose rightmost characters are returned.

nCharacters

Specifies the number of characters returned from the character expression. Right( ) returns the entire character expression if nCharacters is greater than the length of cExpression. Right( ) returns an empty string if nCharacters is negative or 0.

**Returns**

Character

## 16.11.10 StrExtract( )

A Visual FoxPro function which retrieves a string between two delimiters.

**Syntax**

```
StrExtract(cSearchExpression, cBeginDelim, [cEndDelim, [nOccurrence, [nFlag]]])
```

**Argument**

cSearchExpression

Specifies the string to search (If cBeginDelim is an empty string, the search is conducted from the beginning of cSearchExpression to the first occurrence of cEndDelim.)

cBeginDelim

© Welkin Systems Limited

Specifies the character that delimits the beginning of cSearchExpression.

cEndDelim

Specifies the character that delimits the end of cSearchExpression (If cEndDelim is an empty string, StrExtract( ) returns a string from nOccurrence of cBeginDelim to the end of cSearchExpression.)

nOccurrence

Specifies at which occurrence of cBeginDelim in cSearchExpression to start the extraction.

nFlag

Specify the type of controls placed on the search:

| nFlag (additive) | Description |
|---|---|
| 1 | Case-insensitive search |
| 2 | End delimiter not required. Specifies that a search, which finds no occurrence of cEndDelim, returns the contents of cSearchExpression from the cBeginDelim location. |
| 4 | Include the delimiters in the returned expression |

The default is a case sensitive search in which delimiters must be found (no nFlag value).

**Returns**

Character

**Example**

```
StrExtract('ABCDaATC','B','A')
```

Returns 'CDa'

```
StrExtract('ABCDaATC','b','a',1,1)
```

Returns 'CD'

```
StrExtract('ABCDaATC','a','c',2,1)
```

Returns 'AT'

```
StrExtract('ABCDaATC','b','k',1,3)
```

Returns 'CDaATC'

## 16.11.1 SubStr( )

A Visual FoxPro built-in function which returns a character string from the given character expression.

**Syntax**

```
SubStr(cExpression,nStartPosition,nCharactersReturned)
```

© Welkin Systems Limited

**Argument**

cExpression

Specifies the character expression from which the character string is returned.

StartPosition

Specifies the position in the character expression cExpression from where the character string is returned. The first character of cExpression is position 1.

nCharactersReturned

Specifies the number of characters to return from cExpression. If you omit nCharactersReturned, characters are returned until the end of the character expression is reached.

**Returns**

Character

### 16.11.11 Trim( )

A Visual FoxPro built-in function which removes all trailing spaces or parsing characters from the specified character expression.

**Syntax**

Trim(cExpression, nFlags, cParseChar, cParseChar2, ...)

**Argument**

cExpression

Specifies an expression of character to remove all trailing spaces.

nFlags

Specifies if trimming is case-sensitive when one or more parse characters (cParseChar, cParseChar2, … are included. Trimming is case-sensitive if nFlags is zero or is omitted. Trimming is case-insensitive if nFlags 1.

cParseChar, cParseChar2, ...

Specifies one or more character strings that are trimmed from the end of cExpression. If cParseChar isn't included, then trailing spaces. The maximum number of strings permitted in cParseChar is 23.

**Returns**

Character

**Examples**

```
Trim('ABC/..  ')
```

Returns 'ABC/..'

© Welkin Systems Limited

```
Trim('ABC/..  ',[ ],[.],[/])
```

Returns 'ABC'

## 16.12 List Functions

List string is an import tool for processing race histories. RaceMate comes with a set of powerful list functions which helps you effectively process list strings.

**>>> Subtopics of this section <<<**

- ListAscItem( )
- ListAscVal( )
- ListAverage( )
- ListCol( )
- ListColVal( )
- ListCommon( )
- ListCount( )
- ListDescItem( )
- ListDescVal( )
- ListFilter( )
- ListItem( )
- ListSort( )
- ListSortDesc( )
- ListsortDescVal( )
- ListSortVal( )
- ListStdDev( )
- ListSum( )
- ListVal( )
- RListItem( )
- RListVal( )

### 16.12.1 ListAscItem( )

The function sorts an item string in ascending order and returns a specific item. If the items of the item string contain sub-items, you may specify a sub-item as the sorting key. The function can return an item, a sub-item, or the ordinal of an item.

**Syntax**

```
ListAscItem(cValueExpr,nItemNumber,cSeparator,nSortSubItem,cSubItemSeparator,cReturnType)
```

Or

```
ListAscItem(cValueExpr,nItemNumber,cSeparator,cSortExpr,cSubItemSeparator,cReturnType)
```

| Argument | Description | Example |
|---|---|---|
| cValuleExpr | The item string or an expression returning an item string | '12:45/6:00/9:40' |
| nItemNumber | An ordinal number specifying which item, after sorting, to be returned | 3 |

© Welkin Systems Limited

| | | |
|---|---|---|
| cSeparator | Item separator, can be more than 1 character (default to ',') | '/' |
| nSortSubItem | An ordinal number specifying which sub-item to be sorted. If omitted, the whole item will be sorted | 2 |
| cSortExpr | Use the value returned by ListCol( ) and ListColVal( ) to determine the order. Please see the examples below for details | 'ListColVal(1)+ListColVal(2) |
| cSubItemSeparator | Sub-item separator, can be more than 1 character (default to '-') | ':' |
| cReturnType | 'I': Returns the ordinal number, before sorting, of the item. 'R': Returns the whole item Omitted or any other character: returns the sub-item | 'I' |

**Returns**

Character

**Example**

```
ListAscItem('Orange,Apple,Grape,Pear',2)
```

Returns 'Grape'. After sorting, the item string becomes: 'Apple,Grape,Orange,Pear'，The 2nd item is 'Grape'.

```
ListAscItem('12:45/6:00/9:40',3,'/',2,':','I')
```

Returns 1. The 2nd sub-items of the item string are '45', '00', and '40' respectively. After sorting on the 2nd sub-items, the item string becomes '6:00/9:40/12:45', the 3rd item is '12:45', the ordinal number of this item before sorting is 1 (cReturnType='I').

```
ListAscItem('5-4,23-15,3-26',1,,2)
```

Returns '15'. The 2nd sub-items of the item string are '4', '15' and '26' respectively. (Please note that when comparing strings, '4' is greater than '15' since '4' > '1'. When comparing numbers, 4 is smaller 15) Therefore, after sorting, the item string becomes '23-15,5-4,3-26' and the function returns the sub-item '15' of the first item (sub-item is returned if argument cReturnType is omitted).

```
ListAscItem('5-4,23-15,3-26',1,,2,,'R')
```

Returns '23-15'. This example is similar to the example above. The only difference is the whole item is returned (cReturnType='R') instead of the index index.

In the example view in the figure below, the column "WeightFp" returns an item string of both the weight and the finishing position of 8 past starts of each horse by scanning the history only once. Columns "Lightest" and "Best" utilize the ListAscItem( ) function to sort the list by weight and finishing position respectively and returns the

© Welkin Systems Limited

minimum values of the data. Instead of doing it this way, you may apply the functions HrMinimum('HrWeight( )',8) and HrMinimum('HrFP( )',8) to get the same results. However, each of these functions scans the history once. Thus the history will be scanned twice. The speed is slower than that of the view in this example.

| Column | Formula | Data Type |
|---|---|---|
| Horse | (Built-in column) | |
| WeightFp | HrHistoryList("Str(HrWeight( ),3)+'-'+Str(HrFP( ),2)",8) | Text (56) |
| Lightest | ListAscItem(Column(' WeightFp '),1,,1) | Text (3) |
| Best | ListAscItem(Column(' WeightFp '),1,,2) | Text (2) |

| Horse | WeightFp | Lightest | Best |
|---|---|---|---|
| Lombard | 130- 2,124-11,132-14,133-13,128-13,133- 8,115- 2,119- 3 | 115 | 2 |
| Amity | 119-11,115- 4,115- 3,133- 1,123-10,132- 2,116- 7,133- 1 | 115 | 1 |
| Handsome Fortune | 130-11,119- 6,121- 6,124- 7,109- 7,115-12,124-11,126-11 | 109 | 6 |
| Darwin | 116-14,123- 9,129- 5,121-10,122- 3,119-11,123-11 | 116 | 3 |
| Happy Life | 123- 7,128- 4,118- 8 | 118 | 4 |
| Hennessy Star | 116- 8,116- 6,116- 7,117-11 | 116 | 6 |
| Arabica | 117-10,126- 8,122- 4,121-11,117- 3,120-11,114- 9,112-11 | 112 | 3 |
| Fortune Sallian | 124- 4,134- 6,116- 9,118- 3,125- 8,115- 9,132- 3,128- 9 | 115 | 3 |
| Famous Brand | 126-11,126-12,127- 6 | 126 | 6 |
| Perfect Sport | 134- 3,126- 7,111- 1,129- 6,120-10,123-14,118-13,116- 3 | 111 | 1 |
| Winning Mascot | 129-11,126- 5,131-11,125-10,122-13,113- 9,116-10,113- 9 | 113 | 5 |
| Tar 'n' Feather | 124-12,127- 8 | 124 | 8 |

```
ListAscItem('5-4,6-1,2-6',1,,'ListColVal(1)+ListColVal(2)')
```

Returns '6-1'. The argument 'ListColVal(1)+ListColVal(2)' asks the system to sort on the numeric sum of the 1st and 2nd sub-items:

| Item | ListColVal(1)+ListColVal(2) |
|---|---|
| '5-4' | 5 + 4 = 9 |
| '6-1' | 6 + 1 = 7 |
| '2-6' | 2 + 6 = 8 |

Since 7<8<9, after sorting the item string becomes '6-1,2-6,5-4'. The function returns the first item '6-1'.

```
ListAscItem('5-4,6-12,2-6',1,,'ListCol(1)+ListCol(2)')
```

Returns '6-1'. The argument 'ListCol(1)+ListCol(2)' ask the system to sort on the concatenation of the 1st and the 2nd sub-items:

| Item | ListCol(2)+ListCol(1) |
|---|---|
| '5-4' | '4' + '5' = '44' |
| '6-12' | '12' + '6' = '126' |
| '2-6' | '6' + '2' = '62' |

Since '126'<'44'<'62' (note: '126'<'44' because '1'<'4'), the item string after sorting becomes '6-12,5-4,2-6'. The function returns the first item '6-12'.

**See also**

© Welkin Systems Limited

ListDescVal( ) | ListDescVal( ) | ListAscVal( ) | ListItem( ) | ListVal( ) | ListSum( ) | ListAverage( ) | ListFilter( ) |

## 16.12.2 ListAscVal( )

This function is almost identical to ListAscItem( ) except that, in this function, items (or sub-item), which is to be sorted, will be converted to numbers before sorting. Please see ListAscItem( ) for other function details.

### Syntax

```
ListAscVal(cValueExpr,nItemNumber,cSeparator,nSortSubItem,cSubItemSeparator,cReturnType)
```

Or

```
ListAscVal(cValueExpr,nItemNumber,cSeparator,cSortExpr,cSubItemSeparator,cReturnType)
```

### Returns

Character

### Example

```
ListAscVal("8,4,12",1)
```

Returns '4'◦ Compare ListAscItem("8,4,12",1), which will return '12'. It is because in text comparison, '12' is smaller than '4' (since '1' < '4').

### See also

ListDescVal( ) | ListAscItem( ) | ListDescItem( ) | ListItem( ) | ListVal( ) | ListSum( ) | ListAverage( ) | ListFilter( ) |

## 16.12.3 ListAverage( )

This function returns the average of the values of all items in an item string. During the calculation, the system will utilize the Val( ) function to convert items to numbers.

### Syntax

```
ListAverage(cValueExpr,cSeparator)
```

| | |
|---|---|
| cValueExpr | Item string (or function returning an item string) |
| cSeparator | Item separator (can be more than 1 character). Default to ',' |

### Returns

Numeric

### Example

```
ListAverage("8,4,3")
```

Returns 5

© Welkin Systems Limited

**See also**

ListAscItem( ) | ListAscVal( ) | ListDescItem( ) | ListDescVal( ) | ListItem( ) | ListVal ( ) | ListSum( ) | ListCount( ) | ListFilter( ) | ListStdDev( )

## 16.12.4 ListCol( )

This function cannot be used independently. They can only exist in the arguments of the functions ListAscItem( ), ListAscVal( ), ListDescItem( ), ListDescVal( ), and ListListFilter( ).

## 16.12.5 ListColVal( )

This function cannot be used independently. They can only exist in the arguments of the functions ListAscItem( ), ListAscVal( ), ListDescItem( ), ListDescVal( ), and ListListFilter( ).

## 16.12.6 ListCommon( )

This function returns the common items in 2 item strings.

**Syntax**

`ListCommon(cValueExpr1, cValueExpr2,cSeparator)`

| | |
|---|---|
| cValueExpr1 | First item string |
| cValueExpr2 | Second item string |
| cSeparator | Item separator (can be more than 1 character). Default to ',' |

**Returns**

Character

**Example**

```
ListCommon("A,B,C","D,C,B")
```

Returns "C,B"

**See also**

ListAscItem( ) | ListAscVal( ) | ListDescItem( ) | ListDescVal( ) | ListItem( ) | ListVal ( ) | ListSum( ) | ListFilter( ) | ListAverage( ) |

## 16.12.7 ListCount( )

This function returns the number of items in an item string

© Welkin Systems Limited

### Syntax

```
ListCount(cValueExpr,cSeparator)
```

| | |
|---|---|
| cValueExpr | Item string |
| cSeparator | Item separator (can be more than 1 character). Default to ',' |

### Returns

Numeric

### Example

```
ListCount("8,4,3")
```

Returns 3

### See also

ListAscItem( ) | ListAscVal( ) | ListDescItem( ) | ListDescVal( ) | ListItem( ) | ListVal ( ) | ListSum( ) | ListAverage( ) | ListFilter( ) |

## 16.12.8 ListDescItem( )

The function is similar to the ListAscItem( ) function except that items in the item string will be sorted in descending order instead of ascending order. See ListAscItem( ) for details

### Syntax

```
ListDescItem(cValueExpr,nItemNumber,cSeparator,nSortSubItem,cSubItemSeparator,cReturnType)
```

Or

```
ListDescItem(cValueExpr,nItemNumber,cSeparator,cSortExpr,cSubItemSeparator,cReturnType)
```

### Returns

Character

### Example

```
ListDescItem('Orange,Apple,Grape,Pear',2)
```

Returns 'Orange'. After sorting, the item string becomes: 'Pear,Orange,Grape,Apple'. The 2nd item is 'Orange'.

### See also

ListAscItem( ) | ListDescVal( ) | ListAscVal( ) | ListItem( ) | ListVal( ) | ListSum( ) | ListAverage( ) | ListFilter( ) |

© Welkin Systems Limited

## 16.12.9 ListDescVal( )

This function is almost identical to ListDescItem( ) except that, in this function, items (or sub-item), which is to be sorted, will be converted to numbers for sorting. Please see ListAscItem( ) and ListDescItem( ) for function details.

### Syntax

```
ListDescVal(cValueExpr,nItemNumber,cSeparator,nSortSubItem,cSubItemSeparator,cReturnType)
```

Or

```
ListDescVal(cValueExpr,nItemNumber,cSeparator,cSortExpr,cSubItemSeparator,cReturnType)
```

### Returns

Character

### Example

```
ListDescVal("8,4,12",1)
```

Returns '12'。Compare ListAscItem("8,4,12",1), which will return '8'. It is because to in text text comparison, '12' is smaller than '8' since '1' < '8'.

### See also

ListDescItem( ) | ListAscItem( ) | ListAscVal( ) | ListItem( ) | ListVal( ) | ListSum( ) | ListAverage( ) | ListFilter( ) |

## 16.12.1(ListFilter( )

This function extracts a set of items, according to a user define criteria, from an item string to form a new item string.

### Syntax

```
ListFilter(cValueExpr,cFilterExpr,cReturnExpr,cSeparator,cSubItemSeparator)
```

| Argument | Description |
|---|---|
| cValueExpr | Item string (or MCL function returning an item string) |
| cFilterExpr | A logic expression for filtering items. You may use ListCol( ) and ListColVal( ) to compose the expression. ListCol( ) is referred to an item string or sub-item string while ListColVal( ) is referred to the numeric value of an item string or sub-item string. See examples below for details. |
| cReturnExpr | An expression for forming the return items. You may use ListCol( ) and ListColVal( ) to compose the expression. ListCol ( ) is referred to an item string or sub-item string while ListColVal( ) is referred to the numeric value of an item string or sub-item string. See examples below for details. If the argument is omitted, the whole item will be included in the return string. |
| cSeperator | Item separator (can be more than 1 character). Default to ',' |
| cSubItemSeparator | Sub-item separator (can be more than 1 character). Default |

© Welkin Systems Limited

| | |
|---|---|
| | to '-' |

**Returns**

Character

**Example**

```
ListFilter("8,4,3,12",'ListCol( )>"3"')
```

Returns '8,4'. Among the items in the item string, only '8' and '4' are greater than 3. Please note that in string comparison, '12' is smaller than '3'.

```
ListFilter("8,4,3,12",'ListColVal( )>3')
```

Returns '8,4,12'。If items are converted to numbers, only the third item, 3, is not greater then 3.

```
ListFilter("A-5,J-4,C-13,K-12",'ListCol(1)>"D"')
```

Returns 'J-4,K-12'. ListCol(1) is used for specifying the value criterion for the first sub-item, among the all the first sub-items 'A','J','C', and 'K', only 'J' and 'K' are greater than 'D'. Therefore 'J-4,K-12' is returned.

```
ListFilter("A-5,J-4,C-13,K-12",'ListColVal(2)>10')
```

Returns 'C-13,K-12'. ListColVal(2) is used for specifying the criterion for selecting the 2nd sub-item. The items will be converted to number for before selection is made. Among the four sub-items (5,4,13,12), 13 and 12 are greater than 10, therefore 'C-13, K-12' is returned.

```
ListFilter("A-5,J-4,C-13,K-12",'ListColVal(2)>10','ListCol(1)')
```

Returns 'C,K'. Similar to the example above, ListColVal(2) is used for specifying the criterion for the 2nd sub-item. The third argument ListCol(1) specifies that only the first sub-item is to be included in the returning string. Therefore 'C,K' is returned.

```
ListFilter("A 5 5/J 4 6/C 7 8",'ListColVal(2)+ListColVal(3)>9'
       And ListCol(1)<"D"','ListColVal(2)+ListColVal(3)',"/"," ")
```

Returns '10,15'. This example shows that you may use compound expression for both the filter expression and return value expression. Among the items, 'A 5 5' and 'C 7 8' meet the criteria ListColVal(2)+ListColVal(3)>9 (5+5>9，7+8>9) and ListCol(1) <"D" ("A"<"D"，"C"<"D"). The returning item string is formed by ListColVal(2)+ListColVal (3), i.e., 5+5 and 7+8, hence '10,15'.

## 16.12.1 ListItem( )

The ListItem( ) function retrieves a text item from a string of items separated by ','.

**Syntax**

```
ListItem(cValueExpr,nItemNumber,cSeparator,nItemsReturn)
```

**Returns**

© Welkin Systems Limited

Character

**Examples**

```
ListItem("12,41,8",2)
```

The above function returns characters "41".

```
ListItem(Column('Position'),1)
```

The above function returns the first number item in the string in column 'Position'.

```
ListItem(HrOdds( ),2,' ')
```

The above function returns odds 5-minute before race of the horse by retrieving the 2nd item from an odds string. If HrOdds( ) returns "5.8 6.4 7.1", "6.4" is returned by the above function. Please note that the separator in an odds string is a space instead of a ",".

**See Also**

HrSumCount( ), JrSumCount( ), SrSumCount( ), HrOdds( ), JrOdds( ), SrOdds( ), Column( )

## 16.12.11 ListSort( )

The function sorts an item string in ascending order and returns the sorted string or a specific number of items. If the items of the item string contain sub-items, you may specify a sub-item as the sorting key.

**Syntax**

```
ListSort(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

Or

```
ListSort(cValueExpr,nItemsReturned,cSeparator,cSortExpr,cSubItemSeparator)
```

| Argument | Description | Example |
|---|---|---|
| cValueExpr | The item string or an expression returning an item string | '12:45/6:00/9:40' |
| nItemsReturned | Return how many numbers of items. Default to all items. Default to all items. | 3 |
| cSeparator | Item separator, can be more than 1 character (default to ',') | '/' |
| nSortSubItem | An ordinal number specifying which sub-item to be sorted. If omitted, the whole item will be sorted | 2 |
| cSortExpr | Use the value returned by ListCol( ) and ListColVal( ) to determine the order. Please | 'ListColVal(1)+ListColVal(2) |

© Welkin Systems Limited

| | | |
|---|---|---|
| | see the examples below for details | |
| cSubItemSeparator | Sub-item separator, can be more than 1 character (default to '-') | ':' |

**Returns**

Character

**Example**

```
ListSort('Orange,Apple,Grape,Pear')
```

Returns 'Apple,Grape,Orange,Pear'.

```
ListSort('12:45/6:00/9:40',3,'/',2,':')
```

Returns '6:00/9:40/12:45'. The 2nd sub-items of the item string are '45', '00', and '40' respectively. After sorting on the 2nd sub-items, the item string becomes '6:00/9:40/12:45'.

```
ListSort('5-4,23-15,3-26',1,,2)
```

Returns '23-15,5-4,3-26'. The 2nd sub-items of the item string are '4', '15' and '26' respectively. (Please note that when comparing strings, '4' is greater than '15' since '4' > '1'. When comparing numbers, 4 is smaller 15) Therefore, after sorting, the item string becomes '23-15,5-4,3-26'.

```
ListSort('5-4,6-1,2-6',1,,'ListColVal(1)+ListColVal(2)')
```

Returns '6-1,2-6,5-4'. The argument 'ListColVal(1)+ListColVal(2)' asks the system to sort on the numeric sum of the 1st and 2nd sub-items:

| Item | ListColVal(1)+ListColVal(2) |
|---|---|
| '5-4' | 5 + 4 = 9 |
| '6-1' | 6 + 1 = 7 |
| '2-6' | 2 + 6 = 8 |

Since 7<8<9, after sorting the item string becomes '6-1,2-6,5-4'.

```
ListAscItem('5-4,6-12,2-6',1,,'ListCol(1)+ListCol(2)')
```

Returns '6-12,5-4,2-6'. The argument 'ListCol(1)+ListCol(2)' ask the system to sort on the concatenation of the 1st and the 2nd sub-items:

| Item | ListCol(2)+ListCol(1) |
|---|---|
| '5-4' | '4' + '5' = '44' |
| '6-12' | '12' + '6' = '126' |
| '2-6' | '6' + '2' = '62' |

Since '126'<'44'<'62' (note: '126'<'44' because '1'<'4'), the item string after sorting becomes '6-12,5-4,2-6'.

© Welkin Systems Limited

## 16.12.13 ListSortDesc( )

The function is similar to the ListSort( ) function except that items in the item string will be sorted in descending order instead of ascending order. See ListSort( ) for details

### Syntax

```
ListSortDesc(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

Or

```
ListSortDesc(cValueExpr,nItemsReturned,cSeparator,cSortExpr,cSubItemSeparator)
```

### Returns

Character

### Example

```
ListSortDesc('Orange,Apple,Grape,Pear')
```

Returns 'Pear,Orange,Grape,Apple'.

## 16.12.14 ListsortDescVal( )

This function is almost identical to ListSortDesc( ) except that, in this function, items (or sub-item), which is to be sorted, will be converted to numbers for sorting. Please see ListSort( ) for function details.

### Syntax

```
ListSortDescVal(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

Or

```
ListSortDescVal(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

### Returns

Character

### Example

```
ListSortDescVal("8,4,12",1)
```

Returns '12,8,4'。Compare ListSortDesc("8,4,12",1), which will return '8,4,12'. It is because to in text comparison, '12' is smaller than '8' since '1' < '8'.

## 16.12.15 ListSortVal( )

This function is almost identical to ListSort( ) except that, in this function, items (or sub-item), which is to be sorted, will be converted to numbers before sorting. Please see ListSort( ) for other function details.

### Syntax

© Welkin Systems Limited

```
ListSortVal(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

Or

```
ListSortVal(cValueExpr,nItemsReturned,cSeparator,nSortSubItem,cSubItemSeparator)
```

**Returns**

Character

**Example**

```
ListSortVal("8,4,12",1)
```

Returns '4,8,12'. Compare ListSort("8,4,12",1), which will return '12,4,8'. It is because in text comparison, '12' is smaller than '4' (since '1' < '4').

## 16.12.16 ListStdDev( )

This function returns the standard deviation of values of all items in an item string. During the calculation, the system will utilize the Val( ) function to convert items from text to numbers.

**Syntax**

```
ListStdDev(cValueExpr,cSeparator)
```

| | |
|---|---|
| cValueExpr | Item string (or expression returning an item string |
| cSeparator | Item separator (can be more than 1 character). Default to ',' |

**Returns**

Numeric

**Example**

```
ListSum("8,4,3")
```

Returns 2.16

**See also**

ListAscItem( ) | ListAscVal( ) | ListDescItem( ) | ListDescVal( ) | ListItem( ) | ListVal ( ) | ListAverage( ) | ListCount( ) | ListFilter( ) | ListSum( ) |

## 16.12.17 ListSum( )

This function returns the sum of values of all items in an item string. During the calculation, the system will utilize the Val( ) function to convert items from text to numbers.

**Syntax**

```
ListSum(cValueExpr,cSeparator)
```

© Welkin Systems Limited

| | |
|---|---|
| cValueExpr | Item string (or expression returning an item string |
| cSeparator | Item separator (can be more than 1 character). Default to ',' |

**Returns**

Numeric

**Example**

```
ListSum("8,4,3")
```

Returns 15

**See also**

ListAscItem( ) | ListAscVal( ) | ListDescItem( ) | ListDescVal( ) | ListItem( ) | ListVal ( ) | ListAverage( ) | ListCount( ) | ListFilter( ) | ListStdDev( )

## 16.12.18 ListVal( )

The ListVal( ) function retrieves a numeric value from string of numbers separated by ','.

**Syntax**

```
ListVal(cMCLExpr,nItemNumber,cSeparator,nItemsReturn)
```

**Returns**

Numeric

**Examples**

```
ListVal("12,4,8",2)
```

The above function returns 4.

```
ListVal(Column('WinTk'),1)
```

The above function returns the first number item in the string in column 'WinTk'.

```
ListVal(HrOdds( ),2,' ')
```

The above function returns odds 5-minute before race of the horse by retrieving the 2nd item from an odds string (e.g., "5.8 6.4 7.1"). Please note that the separator in an odds string is a space instead of a ",".

**See Also**

HrSumCount( ), JrSumCount( ), SrSumCount( ), HrOdds( ), JrOdds( ), SrOdds( ), Column( )

© Welkin Systems Limited

## 16.12.19 RListItem( )

The RListItem( ) function retrieves a text item from a string of items separated by ','. The function is almost identical to ListItem( ) except that RListItem( ) counts item numbers from right to left while ListItem( ) counts item number from left to right.

### Syntax

```
RListItem(cValueExpr,nItemNumber,cSeparator,nItemsReturn)
```

### Returns

Character

### Examples

```
RListItem("12,41,8",1)
```

The above function returns the character "8".

```
RListItem(Column('Position'),1)
```

The above function returns the last number item in the string in column 'Position'.

```
RListItem(HrSectionalTime( ),2,' ')
```

The above function returns time of the last section of a race in string format. For example, "23.9" will be returned if HrSectionalTime( ) returns "14.0 22.3 23.6 23.9". Please note that the separator in the string is a space instead of a ",".

### See Also

ListItem( ), RlistVal( ), ListVal( ), HrSumCount( ), JrSumCount( ), SrSumCount( ), HrOdds( ), JrOdds( ), SrOdds( ), Column( )

## 16.12.20 RListVal( )

The RListVal( ) function retrieves a numeric value from string of numbers separated by ','. The function is almost identical to the ListVal( ) function except that RListVal( ) counts item numbers from right to left while ListVal( ) counts item numbers from left to right.

### Syntax

```
RListVal(cMCLExpr,nItemNumber,cSeparator,nItemsReturn)
```

### Returns

Numeric

### Examples

```
RListVal("12,4,8",1)
```

The above function returns 8.

© Welkin Systems Limited

```
RListVal(Column('WinTk'),1)
```

The above function returns the last number item in the string in column 'WinTk'.

```
RListVal(HrSectionalTime( ),1,' ')
```

The above function returns time of the last section of a race. For example, 23.9 will be returned if HrSectionalTime( ) returns "14.0 22.3 23.6 23.9". Please note that the separator in the string is a space instead of a ",".

**See Also**

ListVal( ), ListItem( ), RlistItem( ), HrSumCount( ), JrSumCount( ), SrSumCount( ), HrOdds( ), JrOdds( ), SrOdds( ), Column( )

## 16.13 Conversion Functions

This group of functions transforms values based on the criteria that you set.

**>>> Subtopics of this section <<<**

- Cnv( )
- Conv( )
- DateStr( )
- DToC( )
- Evl( )
- GCnv( )
- GConv( )
- QStr( )
- NVL( )
- SCnv( )
- SConv( )
- SToT( )
- Str( )
- StrDiff( )
- TblAddVal( )
- TblWrite( )
- Transform( )
- TToS( )
- Val( )

### 16.13.1 Cnv( )

The function searches a list of items (cListToBeSearched) for an item (cToSearch) and returns another a string item at the same ordinal position from another list (cListOfReturnItems).

**Syntax**

```
Cnv(eToSearch,cListToBeSearched,cListOfReturnItems,cNullSearchReturn,cSeperator)
```

eToSearch

© Welkin Systems Limited

String to be searched. eToSearch can be of any data type

cListToBeSearched

List of items for searching (matching). If eToSearch is not of character type, the function will convert each member item of cListToBeSearched into the same data type before comparing. Please don't put any space after each comma. Otherwise the space will be regarded as a character is the item.

cListOfReturnItems

List of items for returning. If the nth item in cListToBeSearched matches eToSearch, the nth item of cListOfReturnItems will be returned. If no match is found in cListToBeSearched, the function will return a value as follows:

- IF the number of items in cListOfReturnItems is greater than the number of items in cListToBeSearched, the first excess item of cListToBeSearched will be returned.
- Otherwise, .Null. will be returned.

Please don't put any space after each comma. Otherwise the space will be regarded as a character is the item.

cNullSearchReturn

If a .Null. value is passed into eToSearch, the function will return the value of cNullSearchReturn. However, if cNullSearchReturn is empty, .NULL. will be returned.

cSeparator

The default list item separator is comma (,), you may change the separator by passing a string into cSeparator.

**Returns**

Character

**Examples**

```
Cnv('b','a,b,c','dd,ee,ff')
```

String "ee" will be returned.

```
Cnv('x','a,b,c','dd,ee,ff')
```

.Null. will be returned.

```
Cnv('x','a,b,c','dd,ee,ff,gg')
```

String "gg" will be returned.

```
Cnv(.Null.,'a,b,c','dd,ee,ff','hh')
```

String "hh" will be returned.

Cnv('RT','LF;RT','Left("Many",2);right("Many",2)',,';')

String 'Right("Many",2)' will be returned.

```
Cnv(y,'10,15,35','dd,ee,ff')
```

If y=10, String "dd" will be returned.

© Welkin Systems Limited

```
Cnv(y,'.T.,.F.','12,23','35')
```

IF y is true, the function returns "12". If y is false the function returns "23". If y is null, "35" will be returned.

```
Cnv(HrDate( ),'{^2000/1/1},{^2000/1/7},{^2000/1/14}','13,39,12')
```

If HrDate( ) is 14th Jan 2000, String "12" will be returned.

**See also**

SCnv( ) | GCnv( ) | ListItem( ) | RListItem( ) | Conv( ) | SConv( ) | GConv( ) | IIF( )

## 16.13.2 Conv( )

The Conv( ) function converts a value to another value according to a lookup table. A lookup table is consisted of 2 columns. The first column is a Key Value column in which a set of original values is stored. The second column is the returned value column in which the values being converted to are stored. A simple lookup table "FPconv" is shown below:

| KeyVal | RetVal |
|---|---|
| 1 | 10 |
| 2 | 5 |
| 3 | 4 |

According to the table, 1 will be converted to 10 and 2 will be converted to 5, and so on. A default returned value could be specified if the value to be converted is not found in the KeyVal column. If no default returned value is specified, .NULL. will be returned.

**Syntax**

```
Conv(cLookupTable, KeyVal, DefaultValue)
```

**Returns**

The data type of the returned value will be identical to that of the RetVal column.

**Example**

```
Conv("FPconv",HrFP( ),0)
```

The above function converts the finishing position according to the conversion table "FPconv".

## 16.13.3 DateStr( )

Converts a date value into a string in a specific format.

**Syntax**

© Welkin Systems Limited

DateStr(dDate, cFormat)

**Argument**

dDate

Date to be converted

cFormat

'ymd' year in 4 digits, month in 2 digits, date in 2 digits

'mdy' month in 2 digits, date in 2 digits, year in 4 digits

'dmy' date in 2 digits, month in 2 digits, year in 4 digits

**Returns**

Character (10 digits)

**Example**

For the race on the first of October in 2009:

| Function | Return Value |
|---|---|
| DateStr(HrDate(), 'ymd') | '2009/10/01' |
| DateStr(HrDate(), 'dmy') | '01/10/2009' |
| DateStr(HrDate(), 'mdy') | '10/01/2009' |

## 16.13.4 DToC( )

The DToC( ) function is a Visual FoxPro function which returns a Character-type date from a Date expression.

**Syntax**

`DToC(dExpression)`

dExpression

Specifies a Date-type value or MCL expression which DToC( ) returns a Character-type date.

**Returns**

Character (width=10)

## 16.13.5 Evl( )

EVL( ) is a Visual FoxPro function. It converts an empty value to another value. Visual FoxPro regards the following as empty values.

- Zero (0)
- False
- Any combination of Space, Chr(9), Chr(13), Chr(10)
- Empty date, i.e., { / / }

© Welkin Systems Limited

You can think of EVL(eExpression1,eExpression2) as shorthand for:

IIF(Empty(eExpression1),eExpression1,eExpression2)

**Syntax**

```
EVL(eExpression1,eExpression2)
```

## 16.13.6 GCnv( )

The function searches a list of items (cListToBeSearched) for an item equal to or greater than the first argument (cToSearch) and returns another a string item at the same ordinal position from another list (cListOfReturnItems). Search will be carried out from left to right. Once an item equal to or greater than cToSearch is found, all the following items will be ignore, the corresponding item in cListOfReturnItems will be returned.

**Syntax**

```
GCnv(eToSearch,cListToBeSearched,cListOfReturnItems,cNullSearchReturn,cSeperator)
```

eToSearch

String to be searched. eToSearch can be of any data type

cListToBeSearched

List of items for searching (matching). The items in the list must be **from small to large**. If eToSearch is not of character type, the function will convert each member item of cListToBeSearched into the same data type before comparing. Please don't put any space after each comma. Otherwise the space will be regarded as a character of the item.

cListOfReturnItems

List of items for returning. If the nth item in cListToBeSearched matches eToSearch, the nth item of cListOfReturnItems will be returned. If no in cListToBeSearch is equal to or greater than eToSearch, the function will return a value as follows:

- IF the items in cListOfReturnItems are more than that in cListToBeSearched, the first excess item of cListToBeSearched will be returned.
- Otherwise, .Null. will be returned.

Please don't put any space after each comma. Otherwise the space will be regarded as a character of the item.

cNullSearchReturn

If a .Null. value is passed into eToSearch, the function will return the value of cNullSearchReturn. However, if cNullSearchReturn is empty, .NULL. will be returned.

cSeparator

The default list item separator is comma (,), you may change the separator by passing a string into cSeparator.

**Returns**

Character

© Welkin Systems Limited

**Examples**

```
GCnv('b','a,b,c','dd,ee,ff')
```

String "ee" will be returned.

```
GCnv('j','c,k,q','dd,ee,ff')
```

String "ee" will be returned.

```
GCnv('y','s,t,x','dd,ee,ff')
```

.Null. will be returned.

```
GCnv('y','s,t,x','dd,ee,ff,gg')
```

String "gg" will be returned.

```
GCnv(.Null.,'a,b,c','dd,ee,ff','hh')
```

String "hh" will be returned.

```
GCnv('RA','LT;RF','Left("Many",2);Right("Many",2)',,';')
```

String 'Right("Many",2)' will be returned.

```
GCnv(y,'3,15,35','dd,ee,ff')
```

If y=10, String "ee" will be returned.

```
GCnv(HrDate( ),'{^2000/1/2},{^2000/1/8},{^2000/1/14}','13,39,12')
```

If HrDate( ) returns 1st Jan 2000, String "13" will be returned.

**See also**

Cnv( ) | SCnv( ) | ListItem( ) | RListItem( ) | Conv( ) | SConv( ) | GConv( ) | IIF( )

## 16.13.7 GConv( )

The GConv( ) function is identical to the Conv( ) function except that if the lookup value is not found in the KeyVal column, the system will search a closest KeyVal which is greater than the lookup value and return the corresponding value in the RetVal column. If no greater value can be found, the value passed to the DefaultVal argument will be returned. If no value is passed through the DefaultVal argument, .NULL. will be returned.

| KeyVal | RetVal |
|---|---|
| 1 | 10 |
| 2 | 5 |
| 3 | 4 |

According to the conversion table above, similar to Conv( ), 1 will be converted to 10

© Welkin Systems Limited

and 2 will be converted to 5. In the KeyVal column, 2.5 is not found. However, 3 is the closest value that is greater than 2.5. If 2.5 is being looked up, 4 (RetVal of 3) will be returned. Similarly, if 0 is being looked up, 10 (RetVal of 1) will be returned. If 8 is being looked up, .NULL. will be returned (in case no DefaultVal is given).

#### Syntax

```
GConv(cLookupTable, KeyVal, DefaultValue)
```

#### Returns

The data type of the returned value will be identical to that of the RetVal column.

#### Example

```
GConv("FPconv",HrFP( ),0)
```

The above function converts the finishing position according to the conversion table "FPconv".

### 16.13.8 QStr( )

The QStr( ) function converts a value to a character string in proper quoted formats. The value can be of any data type.

| Value | Proper format | Example of string returned |
|---|---|---|
| Numeric | No quotes | 18 |
| Text or Character or String | Quoted with one of the following pair of quotes:<br>" "<br>' '<br>[ ] | "18"<br>'18'<br>[18] |
| Date | Quoted with {^ and } in format {^yyyy/mm/dd}. The Year iV Month - Day sequence will be followed | {^2001/03/12} |
| Logical | .T. for true, .F. for false | .T. |

#### Syntax

```
QStr(Value)
```

#### Returns

Character (proper quotes, if any, will be included)

#### Example

```
HrAverage('HrRating( )',,'For HrDate( )>'+QStr(HrDate( )-365))
```

© Welkin Systems Limited

## 16.13.9 NVL( )

NVL( ) is a Visual FoxPro function. It provides a way to keep nulls from propagating without having to use IsNull( ).

You can think of NVL(eExpression1,eExpression2) as shorthand for:

IIF(IsNull(eExpression1),eExpression1,eExpression2)

### Syntax

```
NVL(eExpression1,eExpression2)
```

## 16.13.10 SCnv( )

The function searches a list of items (cListToBeSearched) for an item equal to or smaller than the first argument (cToSearch) and returns another a string item at the same ordinal position from another list (cListOfReturnItems). Search will be carried out from left to right. Once an item equal to or smaller than cToSearch is found, all the following items will be ignore, the corresponding item in cListOfReturnItems will be returned.

### Syntax

```
SCnv(eToSearch,cListToBeSearched,cListOfReturnItems,cNullSearchReturn,cSeperator)
```

eToSearch

String to be searched. eToSearch can be of any data type

cListToBeSearched

List of items for searching (matching). If eToSearch is not of character type, the function will convert each member item of cListToBeSearched into the same data type before comparing. Please don't put any space after each comma. Otherwise the space will be regarded as a character of the item.

cListOfReturnItems

List of items for returning. The items in the list must be **from large to small**. If the nth item in cListToBeSearched matches eToSearch, the nth item of cListOfReturnItems will be returned. If no in cListToBeSearch is equal to or smaller than eToSearch, the function will return a value as follows:

- IF the items in cListOfReturnItems are more than that in cListToBeSearched, the first excess item of cListToBeSearched will be returned.
- Otherwise, .Null. will be returned.

Please don't put any space after each comma. Otherwise the space will be regarded as a character of the item.

cNullSearchReturn

If a .Null. value is passed into eToSearch, the function will return the value of cNullSearchReturn. However, if cNullSearchReturn is empty, .NULL. will be returned.

cSeparator

The default list item separator is comma (,), you may change the separator by passing a string into cSeparator.

© Welkin Systems Limited

**Returns**

Character

**Examples**

```
SCnv('b','a,b,c','dd,ee,ff')
```

String "dd" will be returned.

```
SCnv('j','q,k,c','dd,ee,ff')
```

String "ff" will be returned.

```
SCnv('a','x,t,s','dd,ee,ff')
```

.Null. will be returned.

```
SCnv('a','x,t,s','dd,ee,ff,gg')
```

String "gg" will be returned.

```
SCnv(.Null.,'c,b,a','dd,ee,ff','hh')
```

String "hh" will be returned.

```
SCnv('RT','RF;LT','Left("Many",2);Right("Many",2)',,';')
```

String 'Right("Many",2)' will be returned.

```
SCnv(y,'35,15,3','dd,ee,ff')
```

If y=10, String "ff" will be returned.

```
SCnv(HrDate( ),'{^2000/1/21},{^2000/1/18},{^2000/1/14}','13,39,12')
```

If HrDate( ) is 30th Jan 2000, String "13" will be returned.

**See also**

Cnv( ) | GCnv( ) | ListItem( ) | RListItem( ) | Conv( ) | SConv( ) | GConv( ) | IIF( )

## 16.13.1 SConv( )

The SConv( ) function is identical to the Conv( ) function except that if the lookup value is not found in the KeyVal column, the system will search a closest KeyVal which is smaller than the lookup value and return the corresponding value in the RetVal column. If no smaller value can be found, the value passed to the DefaultVal argument will be returned. If no value is passed through the DefaultVal argument, .NULL. will be returned.

| KeyVal | RetVal |
|---|---|
| 1 | 10 |
| 2 | 5 |

© Welkin Systems Limited

| | |
|---|---|
| 3 | 4 |

According to the conversion table above, similar to Conv( ), 1 will be converted to 10 and 2 will be converted to 5. In the KeyVal column, 2.5 is not found. However, 2 is the closest value that is smaller than 2.5. If 2.5 is being looked up, 5 (RetVal of 2) will be returned. Similarly, if 7 is being looked up, 4 (RetVal of 3) will be returned. If 0 is being looked up, .NULL. will be returned (in case no DefaultVal is given).

**Syntax**

```
SConv(cLookupTable, KeyVal, DefaultValue)
```

**Returns**

The data type of the returned value will be identical to that of the RetVal column.

**Example**

```
SConv("FPconv",HrFP( ),0)
```

The above function converts the finishing position according to the conversion table "FPconv".

### 16.13.12 SToT( )

The SToT( ) function converts a number of seconds to a time string.

e.g. "1.13.41".

**Syntax**

```
SToT(nExpression)
```

nExpression

Specifies the numeric value of seconds.

**Returns**

Character (Width=7)

**Example**

```
SToT(243.2/3)
```

The above function will return the string: "1.21.07" for 1 minute 21.07 seconds.

### 16.13.13 Str( )

The Str( ) function is a Visual FoxPro function which returns the character equivalent of a specified numeric expression.

**Syntax**

© Welkin Systems Limited

```
STR(nExpression,nLength,nDecimalPlaces)
```

nExpression

Specifies the numeric expression STR( ) evaluates.

nLength

Specifies the length of the character string STR( ) returns. The length includes one character for the decimal point and one character for each digit to the right of the decimal point.

STR( ) pads the character string it returns with leading spaces if you specify a length larger than the number of digits to the left of the decimal point. STR( ) returns a string of asterisks, indicating numeric overflow, if you specify a length less than the number of digits to the left of the decimal point. If nLength isn't included, the length of the character string defaults to 10 characters.

nDecimalPlaces

Specifies the number of decimal places in the character string STR( ) returns. You must include nLength to specify the number of decimal places. If you specify fewer decimal places than are in nExpression, the return value is rounded. If nDecimalPlaces isn't included, the number of decimal places defaults to zero.

**Returns**

Character

## 16.13.14 StrDiff( )

The function is returns a character string as the result of the difference between 2 numeric values.

**Syntax**

```
StrDiff(nValue1,nValue2)
```

- The function will return nValue1-nValue2 and convert it into a string. You will find a "+" or "-" at the returned string representing positive or negative of the result.
- If only one value is passed into the function, it will convert this value into a string and add a "+" or "-" at the returned string representing positive or negative of the result.

**Returns**

Character

## 16.13.15 TblAddVal( )

Add a value to the second column of a conversion table. This function is similar to

```
TblWrite(cLookupTable,vKeyVal,Conv(cLookupTable,vKeyVal)+vValueToAdd,vReturnVal,lNotRecalcCond
```

Please note that conversion table name is case-sensitive.

© Welkin Systems Limited

**Syntax**

```
TblAddVal(cLookupTable,vKeyVal,vValueToAdd,vReturnVal,lNotRecalcCond )
```

| cLookupTable | Name of conversion table |
|---|---|
| vKeyVal | KeyVal identifying record of conversion table to be written |
| vValueToAdd | Value to be added to the conversion table |
| vReturnVal | Return value. If vReturnVal is omitted, value of the argument vValueToAdd will be returned. If .T. is passed to vReturnVal, the final value of the RetVal (second) column of the conversion table will be returned. |
| lNotRecalcCond | If lNotrecalcCond is satisfied, TblAddVal( ) will not be re-excuted when you push the "Recalc" button |

**Returns**

Data type identical to that of vReturnVal

**Example**

```
TblAddVal('PlaceCount',HrStartID( ),iif(HrFP( )<=3,1,0),' ',.T.)
```

**See Also**

TblWrite( )

## 16.13.16 TblWrite( )

Write a value into the value column, the second column, of a conversion table. Please note that conversion table is case-sensitive.

**Syntax**

```
TblWrite(cLookupTable,vKeyVal,vWriteVal,vReturnVal,lNotRecalcCond )
```

| cLookupTable | Name of conversion table |
|---|---|
| vKeyVal | KeyVal identifying record of conversion table to be written |
| vWriteVal | Value to be written to the conversion table |
| vReturnVal | Reutrn value. If vReturnVal is omitted, vWriteVal will be returned. |
| lNotRecalcCond | If lNotRecalcCond is satisfied, TblWrite( ) will not be re-executed when you push the "Recalc" button |

**Returns**

Data type identical to that of vReturnVal

**Example**

```
TblWrite('RaceRemark',HrStartID( ),Column('Remark'),' ')
```

**See Also**

© Welkin Systems Limited

TblAddVal( )

### 16.13.17 Transform( )

Transform( ) is a Visual FoxPro function which converts a value of any data type to a character string. Transform( ) can be used in the short form Tran( ).

#### Syntax

`Transform(eExpression)`

#### Returns

Character

#### Examples

| eExpression | Example | Example of string returned |
|---|---|---|
| Numeric | `Transform(HrBodyweight( ))` | 1018 |
| Text | `Transform(HrDistance( ))` | 1200 |
| Date | `Transform(HrDate( ))` | 13/03/2001 |
| Logical | `Transform(IsNull(Column('Performance')))` | .T. |
| .NULL. | `Transform(.NULL.)` | .NULL. |

### 16.13.18 TToS( )

Converts a time string into number of seconds

#### Syntax

`TToS(cExpression)`

cExpression is the string to be converted

#### Returns

Numeric

#### Example

`TToT("1:24.23")`

The function converts "1:24.23" into 84.23

© Welkin Systems Limited

### 16.13.19Val( )

The Val( ) function is a Visual FoxPro function which returns a numeric value from a character expression composed of numbers. Val( ) returns the numbers in the character expression from left to right until a non-numeric character is encountered (leading blanks are ignored). Val( ) returns 0 if the first character of the character expression isn't a number or a plus sign (+) or minus sign (-).

#### Syntax

```
Val(cExpression)
```

cExpression

Specifies a character expression composed of numbers.

#### Returns

Numeric

## 16.14 Multimedia & Internet Functions

This group of functions transforms values based on the criteria that you set.

#### >>> Subtopics of this section <<<

- HrBTVideo( ), JrBTVideo( ), SrBTVideo( )
- HrPostRace( ), JrPostRace( ), SrPostRace( )
- HrPreRace( ), JrPreRace( ), SrPreRace( )
- HrPosPhoto( ), JrPosPhoto( ), SrPosPhoto( )
- HrRaceReplay( ), JrRaceReplay( ), SrRaceReplay( )
- HrRacingFocus( ), JrRacingFocus( ), SrRacingFocus( )

### 16.14.1 HrBTVideo( ), JrBTVideo( ), SrBTVideo( )

The functions provide you with a link for double-clicking to

1. open the Jockey Club barrier trial videos
2. open the barrier trail form if [Alt] key is held down simultaneously

#### Syntax

```
HrBTVideo(nBackBarrierTrials,vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrBTVideo(nBackBarrierTrials,vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrBTVideo(nBackBarrierTrials,vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

#### Argument

nBackBarrierTrials

Pass a number to the first argument to get the date and group of the *nBackBarrierTrials* (0 or 1 for first, 2 for second, etc) barrier trials before the underlying race.

vRetVal

© Welkin Systems Limited

1. If you omit this argument, a character string of 14 characters showing the date and group of the trial will be returned.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (75 characters) for opening the video will be returned. Links to local files will be skipped.

**Returns**

Default to string, e.g., "17/02/2009(5) " for 17/02/2009 group 5. If you pass a value to vRetVal, please see the description for vRetVal above.

**Example**

The following functions will return the latest barrier trial date and group. If you double-click on the date, the barrier trial video will be played.

```
HrBTVideo()
HrBTVideo(1)
```

The following function will give the date and group before the last trial before the underlying race. That is 2nd last trial. This trial may or may not be before the 2nd last race. If you double-click on the date, the barrier trial video will be played.

```
HrBTVideo(2)
```

The following function will give the date and group of the trial before the 2nd last race. If you double-click on the date, the barrier trial video will be played.

```
HrBTVideo(1,,1)
```

The following function will give the latest barrier trial date and group. It will display "Latest BT" on screen. If you double-click on the text, the barrier trial video will be played.

```
HrBTVideo(,"Latest BT")
```

The following function will return the link for getting access to the web resource (something similar to "mms://wm.hkjc.pbase.net/extra1/barrier/20080227/eng_20080227_brts04_500.asf").

```
HrBTVideo(,.Null.)
```

**See Also**

HrBTrial( ), HrPostRace( ), HrPreRace( ), HrPosPhoto( ), HrRaceReplay( ), HrRacingFocus( )

## 16.14.2 HrPostRace( ), JrPostRace( ), SrPostRace( )

The functions provide you with a link for double-clicking to

1. open the "Racing Strategy" videos (in Chinese)
2. open the Race Data form if [Alt] key is held down simultaneously

**Syntax**

```
HrPostRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrPostRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrPostRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

© Welkin Systems Limited

**Argument**

vRetVal

1. If you omit this argument, a character string of 10 characters showing the race date.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (81 characters) for opening the video will be returned. Links to local files will be skipped.

**Returns**

Default to string, e.g., "15/02/2009" for 15/02/2009. If you pass a value to vRetVal, please see the description for vRetVal above.

**Example**

The following function will return the racing date. If you double-click on the date, the corresponding "Post Race" video (in Chinese) will be played.

```
HrPostRace()
```

The following function will give the racing date. It will display "Racing" on screen. If you double-click on the text, the corresponding "Post Race" video (in Chinese) will be played.

```
HrPostRace("Racing")
```

The following function will return the link for getting access to the web resource (something similar to "mms://wm.hkjc.pbase.net/extra1/postrace/20080227/chi_postrace_20080227_04_500.asf").

```
HrPostRace(.Null.)
```

**See Also**

HrBTrial( ), HrBTVideo( ), HrPreRace( ), HrPosPhoto( ), HrRaceReplay( ), HrRacingFocus ( )

## 16.14.3 HrPreRace( ), JrPreRace( ), SrPreRace( )

The functions provide you with a link for double-clicking to

1. open the "Racing to Win" videos
2. open the Race Data form if [Alt] key is held down simultaneously

**Syntax**

```
HrPreRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrPreRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrPreRace(vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

**Argument**

vRetVal

© Welkin Systems Limited

1. If you omit this argument, a character string of 10 characters showing the race date.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (79 characters) for opening the video will be returned. Links to local files will be skipped.

**Returns**

Default to string, e.g., "15/02/2009" for 15/02/2009. If you pass a value to vRetVal, please see the description for vRetVal above.

**Example**

The following function will return the racing date. If you double-click on the date, the corresponding "Racing To Win" video will be played.

```
HrPreRace()
```

The following function will give the racing date. It will display "Racing" on screen. If you double-click on the text, the corresponding "Racing To Win" video will be played.

```
HrPreRace("Racing")
```

The following function will return the link for getting access to the web resource (something similar to "mms://wm.hkjc.pbase.net/extra1/prerace/20080227/eng_prerace_20080227_04_500.asf").

```
HrPreRace(.Null.)
```

**See Also**

HrBTrial( ), HrBTVideo( ), HrPostRace( ), HrPosPhoto( ), HrRaceReplay( ), HrRacingFocus( )

## 16.14.4 HrPosPhoto( ), JrPosPhoto( ), SrPosPhoto( )

The functions provide you with a link for double-clicking to

1. open the "Race Running Position Photos" videos
2. open the Race Data form if [Alt] key is held down simultaneously

**Syntax**

```
HrPosPhoto(vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrPosPhoto(vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrPosPhoto(vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

**Argument**

vRetVal

1. If you omit this argument, a character string of 10 characters showing the race date.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (76 characters) for opening the video will be returned. Links to local files will be skipped.

© Welkin Systems Limited

**Returns**

Default to string, e.g., "15/02/2009" for 15/02/2009. If you pass a value to vRetVal, please see the description for vRetVal above.

**Example**

The following function will return the racing date. If you double-click on the date, the corresponding "Race Running Position Photos" will be opened.

```
HrPosPhoto()
```

The following function will give the racing date. It will display "Positions" on screen. If you double-click on the text, the corresponding "Race Running Position Photos" page will be opened.

```
HrPosPhoto("positions")
```

The following function will return the link for getting access to the web resource (something similar to "http://www.hkjc.com/english/racing/finishphoto.asp?racedate=20080227R4_L.jpg").

```
HrPosPhoto(.Null.)
```

**See Also**

HrBTrial( ), HrBTVideo( ), HrPreRace( ), HrPostRace( ), HrRaceReplay( ), HrRacingFocus( )

## 16.14.5 HrRaceReplay( ), JrRaceReplay( ), SrRaceReplay( )

The functions provide you with a link for double-clicking to

1. open the "Race Replay" videos
2. open the Race Data form if [Alt] key is held down simultaneously

**Syntax**

```
HrRaceReplay(vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrRaceReplay(vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrRaceReplay(vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

**Argument**

vRetVal

1. If you omit this argument, a character string of 10 characters showing the race date.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (74 characters) for opening the video will be returned. Links to local files will be skipped.

**Returns**

Default to string, e.g., "15/02/2009" for 15/02/2009. If you pass a value to vRetVal, please see the description for vRetVal above.

© Welkin Systems Limited

**Example**

The following function will return the racing date. If you double-click on the date, the corresponding "Race Replay" video will be played.

```
HrRaceReplay()
```

The following function will give the racing date. It will display "Racing" on screen. If you double-click on the text, the corresponding "Racing To Win" video will be played.

```
HrRaceReplay("Racing")
```

The following function will return the link for getting access to the web resource (something similar to "mms://wm.hkjc.pbase.net/extra1/replay/20080227/eng_20080227_race04_500.asf").

```
HrRaceReplay(.Null.)
```

**See Also**

HrBTrial( ), HrBTVideo( ), HrPreRace( ), HrPostRace( ), HrPosPhoto( ), HrRacingFocus ( )

## 16.14.6 HrRacingFocus( ), JrRacingFocus( ), SrRacingFocus( )

The functions provide you with a link for double-clicking to

1. open the "Racing Focus" videos (in Chinese)
2. open the Race Calendar form if [Alt] key is held down simultaneously

**Syntax**

```
HrRacingFocus(nSection,vRetVal,nBackRaces,'fCondition',vDefaultVal)
JrRacingFocus(nSection,vRetVal,nBackRaces,'fCondition',vDefaultVal)
SrRacingFocus(nSection,vRetVal,nBackRaces,'fCondition',vDefaultVal)
```

**Argument**

nSection

Specify the section number of the video.

vRetVal

1. If you omit this argument, a character string of 13 characters showing the race date and section of the video.
2. If you pass a value into vRetVal, it will be the return value (can be any type) of the functions.
3. If you pass .Null. into vRetVal, the link (87 characters) for opening the video will be returned. Links to local files will be skipped.

**Returns**

Default to string, e.g., "15/02/2009(3)" for section 3 of 15/02/2009. If you pass a value to vRetVal, please see the description for vRetVal above.

**Example**

The following function will return a date and a section number. If you double-click on

© Welkin Systems Limited

it, the corresponding "Post Race" video (in Chinese) section 1 will be played.

```
HrRacingFocus()
HrRacingFocus(1)
```

The following function will return a date and a section number. If you double-click on it, the corresponding "Post Race" video (in Chinese) section 2 will be played.

```
HrRacingFocus(2)
```

The following function will display "Focus" on screen. If you double-click on the text, the correspon

```
HrRacingFocus("Focus")
```

The following function will return the link for getting access to the web resource (something similar to "mms://wm.hkjc.pbase.net/extra1/RacingFocus/20080227/chi_RacingFocus_20080227_04_500.asf").

```
HrRacingFocus(,.Null.)
```

**See Also**

HrBTrial( ), HrBTVideo( ), HrPreRace( ), HrPostRace( ), HrPosPhoto( ), HrRaceReplay( )

## 16.15 Miscellaneous Functions

Miscellaneous functions not in the basic categories.

**>>> Subtopics of this section <<<**

- Box( )
- Compare( )
- Date( )
- Edition( )
- Empty( )
- Gel( )
- ICase( )
- Iif( )
- InList( )
- IsNull( )
- Join( )
- Lang( )
- Match( )
- Month( )
- Same( )
- Season( )
- Year( )

### 16.15.1 Box( )

Use the Box( ) function to store a value into a RaceMate storage box or return the latest value in the box. RaceMate have 7 storage boxes in total. If you omit the second argument, the function will return the value in the box specified by the first argument. Storage boxes are common in all views. That means once you store a value into a storage box, the value in it can be accessed in any other views. Values storage in a starter view can be accessed in a hist/stat view as well. Values in storage boxes will be kept until you quit RaceMate.

© Welkin Systems Limited

Please note that Box( ) can't be put inside a string. If you really want to use Box( ) in a string, you must precede it with a "&" sign.

### Syntax 1 (outside strings)

```
Box(nBoxNumber,eValueToStore,eArg1,eArg2,eArg3,eArg4,eArg5,eArg6,eArg7,eArg8,eArg9)
```

### Syntax 2 (in strings)

```
&Box(nBoxNumber,eValueToStore,eArg1,eArg2,eArg3,eArg4,eArg5,eArg6,eArg7,eArg8,eArg9)
```

### Argument

nBoxNumber

Indicate which box to be used. RaceMate have 7 storage boxes in total.

eValueToStore

The value to be stored into the box specified by argument 1

eArg1,... eArg9

If eValueToStore is a character string, you may put template symbols @1, @2,... to @9 into eValueToStore. They will be replaced by eArg1, eArg2,... eArg9 in a one-one match basis in the return value. However the @1, @2... @9 template symbols will be kept in the storage box. If eValueToStore is omitted, the replacement will take place when returning the string in the storage box.

### Returns

Data type of the value in the box specified by argument 1

### Example

```
Box(1, HrAge())
```

Put age of horse into storage box number 1

```
Box(4)
```

Return value in storage box number 4

```
Box(3,'HrDaysAgo()+@1',30)
```

Stores 'HrDaysAgo()+@1' into storage box number 3 but returns a string 'HrDaysAgo() +30'.

```
Box(3,,30)
```

If string 'HrDaysAgo()+@1' is stored in the storage box, 'HrDaysAgo()+30' will be returned. However value in the storage box will not be changed by the replacement. Box(3) will still return 'HrDaysAgo()+@1'.

```
ListItem("He is Box(1,'John')|Paul", 1, "|")
```

In the example above, "Box(1, 'John')" will be regarded as a string. The function will return "He is Box(1, 'John')".

© Welkin Systems Limited

```
ListItem("He is &Box(1,'John')|Paul", 1, "|")
```

In the example above "&Box(1, 'John')" will carry out the feature of the Box() function. The function will return "He is John"。

## 16.15.2 Compare( )

The function compares 2 values and return 1, 0, or -1 corresponding to "greater than", "equal to", and "smaller than" respectively. If any of the arguments is .Null., .Null. will be returned.

In Visual FoxPro string comparison using the native operators "<", ">" and "=" will be carried out up to the last character of the right hand side. This behaviour can be a trap for most people. You may keep away from this trap by using Compare( ).

**Using operations >, < and =:**

| Comparison | Returned Value | Remark |
|---|---|---|
| "ABC"="AB" | .T. | In Visual FoxPro string comparison will be carried out up to the last character of the right hand side. Therefore "C" at the left hand side will be ignored |
| "ABC"<"AB" | .F. | |
| "ABC">"AB" | .F. | |
| "AB"="ABC" | .F. | In Visual FoxPro string comparison will be carried out up to the last character of the right hand side. Therefore "C" at the right hand side will be taken into account |
| "AB"<"ABC" | .T. | |
| "AB">"ABC" | .F. | |
| "AB"="AB" | .T. | |
| 3 > 4 | .F. | |
| 3 = 3 | .T. | |
| 4 < 3 | .F. | |
| .Null. > 3 | .Null. | |

**Using the Compare( ) function:**

| Comparison | Returned Value |
|---|---|
| Compare("ABC","AB") | 1 |
| Compare("AB","ABC") | -1 |
| Compare("AB","AB") | 0 |
| Compare(3, 4) | -1 |
| Compare(3, 3) | 0 |
| Compare(4, 3) | 1 |
| Compare(.Null., 3) | .Null. |

### Syntax

Compare(eValue1,eValue2)

### Returns

Numeric

© Welkin Systems Limited

## 16.15.3 Date( )

The Date( ) function is a Visual FoxPro function returns the current system date, which is controlled by the operating system, or creates a year 2000-compliant Date value.

### Syntax 1

```
DATE( )
```

When Date( ) is used without arguments, the current date is return.

### Syntax 2

```
DATE(nYear, nMonth, nDay)
```

nYear

Specifies the year returned in the year 2000-compliant Date value. nYear can be a value from 100 to 9999.

nMonth

Specifies the month returned in the year 2000-compliant Date value. nMonth can be a value from 1 to 12.

nDay

Specifies the day returned in the year 2000-compliant Date value. nDay can be a value from 1 to 31.

### Returns

Date

### See Also

Month( ); HrDate( ), JrDate( ), SrDate( ); Year( )

## 16.15.4 Edition( )

When using syntax 1, the function returns "HK" in HK edition and "MACAU" in Macau edition. When using syntax 2, the function returns the 1st argument in HK edition and the 2nd argument in Macau edition.

### Syntax 1

```
Edition()
```

### Syntax 2

```
Edition(eHkEditionValue,eMacauEditionValue)
```

### Example

© Welkin Systems Limited

```
Edition()
```

Returns "HK" in HK edition and "MACAU" in Macau edition.

```
Edition(3,2)
```

Returns 3 in HK edition and 2 in Macau edition.

### 16.15.5 Empty( )

Empty( ) is a built-in Visual FoxPro function which determines whether an expression evaluates to empty.

The expression you include can be a character, numeric, date, or logical expression. EMPTY( ) returns true (.T.) when expressions evaluate to the following:

| Expression type | Evaluates to |
|---|---|
| Character | The empty string, spaces, tabs, carriage returns, linefeeds, or any combination of these. |
| Numeric | 0 |
| Date | Empty ({}) |
| Logical | False (.F.) |

#### Syntax

```
Empty(eExpr)
```

#### Returns

Logical

### 16.15.6 Gel( )

Returns a string which is a concatenate off all items. Items can be any data type. The maximum number of items is 27. All blanks before or after each item will be trimmed.

#### Syntax

```
Gel(eItem1, eItem2, eItem3,...)
```

#### Returns

Character

#### Examples

```
Gel(' Apple ', 30)
```

Returns 'Apple30'. Please note blanks before and after 'Apple' are trimmed.

```
Gel(HrJockey( ), HrDistance( ))
```

Returns a concatenation of jockey name and race distance, for example, 'D Whyte1400'.

© Welkin Systems Limited

## 16.15.7 lCase( )

This is a Visual FoxPro function which evaluates the results from a list of conditions.

### Syntax

```
ICase( lCondition1, eResult1, lCondition2, eResult2, eOtherwiseResult])
```

### Arguments

lCondition

Specifies a condition as a logical expression to evaluate.

If lCondition evaluates to False (.F.), ICASE( ) continues to evaluate the next condition and returns the corresponding eResult for that condition if it evaluates to True (.T.). If lCondition evaluates to null (.NULL.), Visual FoxPro treats lCondition as if it evaluated to False (.F.).

eResult

Specifies a result to return if lCondition evaluates to True (.T.).

eOtherwiseResult

Contains the result returned if all conditions evaluate to False (.F.).

### Return Value

ICASE( ) returns the first eResult as soon as lCondition evaluates to True (.T.).

If all conditions evaluate to False (.F.), ICASE( ) returns eOtherwiseResult.

If eOtherwiseResult is omitted, and all conditions evaluate to False (.F.), ICASE( ) returns null (.NULL.).

### See Also

IIF( )

## 16.15.8 Iif( )

This is a Visual FoxPro function. It allows you to put equivalent of an IF..ELSE..ENDIF of common programming languages in an MCL formula.

### Syntax

```
Iif(lCondition,eThenResult,eElseResult)
```

| Argument | Value | Meaning |
|---|---|---|
| lCondition | .T. | Returns the value of eThenResult |
| | .F. or .NULL. | Returns the value of eElseResult |
| eThenResult | Any valid expression | If lCondition is true, returns the value of this expression |

© Welkin Systems Limited

| | | |
|---|---|---|
| eElseResult | Any valid expression | If lCondition is false, returns the value of this expression |

## 16.15.9 InList( )

This is a Visual FoxPro function. It determines whether an expression matches another expression in a set of expressions. Here match mean a Visual FoxPro "=" comparison. That means left hand side string can be longer than the right hand side string. See Compare( ) for the rule for matching.

### Syntax

```
InList(eExpression1, eExpression2 , eExpression3 ...)
```

### Example

```
InList(x, 3, 7, 9)
```
The above function will return .T. is x equals 9.

```
InList(x, 'Apples', 'Oranges')
```
The above function will return .T. if x equals 'Apples and Bananas' or 'Apples'. It will return .F. if x equals 'Apple' or 'APPLES'.

### See Also

Compare( ), Match( )

## 16.15.10 IsNull( )

IsNull( ) is a Visual FoxPro function. You need IsNull( ) because .NULL. is never equal to any value, even itself. Nor is it unequal to other values. That is, .NULL.=.NULL. returns .NULL. and so does .NULL.<>.NULL.

The only way to find out if you have a null value is to test with IsNull( ).

### Syntax

```
IsNull(eExpression)
```

### Returns

Logical

## 16.15.11 Join( )

Returns a string which is a concatenate off all items. Items can be any data type. The maximum number of items is 26. (The separator argument will not be counted as an item). All blanks before or after each item will be trimmed.

### Syntax

```
Join(cSeperator, eItem1, eItem2, eItem3,...)
```

© Welkin Systems Limited

**Argument**

cSeperator

A string to be put between 2 consecutive items. Default to a space.

eItem1, eItem2, eItem3,...

Values to be concatenated. The values can be any data type. The maximum number of items is 26.

**Returns**

Character

**Examples**

```
Join( , ' Apple ', 30)
```

Returns 'Apple 30'. Please note blanks before and after 'Apple' are trimmed.

```
Join('-', HrJockey( ), HrDistance( ))
```

Returns a concatenation of jockey name and race distance separating by a hyphen, for example, 'D Whyte-1400'.

## 16.15.12 Lang( )

The function returns the value of the 1st argument in English mode and that of the 2nd argument in Chinese mode.

**Syntax**

```
Lang(eEnglishModeExpr,eChineseModeExpr)
```

**Example**

```
Edition('A','B')
```

Returns 'A' in English mode and 'B' in Chinese Mode.

## 16.15.13 Match( )

The function compares 2 values and if they are equal (.T.) or not (.F.). In Visual FoxPro string comparison using the native operators "=" will be carried out up to the last character of the right hand side. This behaviour can be a trap for most people. You may keep away from this trap by using Match( ).

If there is any of the wildcard characters * and ? in the second argument. The second argument will be treated as a template. "*" matches any number of any characters while "?" matches any single character.

**Using =:**

| Compare | Return | Remark |
|---|---|---|

© Welkin Systems Limited

| | | |
|---|---|---|
| "ABC"="AB" | .T. | In Visual FoxPro string comparison will be carried out up to the last character of the right hand side. Therefore "C" at the left hand side will be ignored |
| "AB"="ABC" | .F. | In Visual FoxPro string comparison will be carried out up to the last character of the right hand side. Therefore "C" at the right hand side will be taken into account |
| "AB"="AB" | .T. | |

**Using Match( ):**

| Comparison | Returned Value |
|---|---|
| Match("ABC","AB") | .F. |
| Match("AB","ABC") | .F. |
| Match("AB","AB") | .T. |
| Match('Ab','ab') | .T. |
| Match('Ab','ab',1) | .F. |
| Match('Abc','a*') | .T. |
| Match('Abc','a*',1) | .F. |
| Match('Abc','A??') | .T. |
| Match('Abc','ABC ') | .T. |
| Match('Abc','ABC ',2) | .F. |
| Match('Abc','Abc ',3) | .F. |

### Syntax

```
Match(cValue1,cValue2,nFlag)
Match(cValue1,cTemplate,nFlag)
```

### Arguments

cValue1, cValue2

Strings to be compared

nFlag

0 or Omit: Trim character strings and upper and lower case character will be treated as the same
1: Require a exact case match
2: Do not trim character strings
The above values are additive.

### Returns

Logical

## 16.15.14 Month( )

Month( ) is a Visual FoxPro function. It returns the number of the month for a given date.

© Welkin Systems Limited

**Syntax**

```
Month(dExpression1)
```

**Returns**

Numeric

**Example**

```
Month({^2003/8/1})
```

Returns 8 for August.

**See Also**

Date( ); Year( ); HrDate( ), JrDate( ), SrDate( )

### 16.15.15 Same( )

The function will return a condition expression. The left hand side of the condition expression is an MCL expression. The right hand side of the condition expression is the value of the MCL expression in current start. The operator of the condition expression is default to "=". The MCL expression must be a simple expression without any "And" or "Or" (can contain + - X / ).

**Syntax**

```
Same(cExpression,cOperator)
```

**Example**

```
Same('HrDistance( )')
```

If the above function is applied to a 1650m start, the function will return

```
'HrDistance( )=1650'
```

```
HrAverage('HrSpeed( )',6,'For '+Same('HrDistance( )'))
```

If the above function is applied to a 1650m start, the function will return

```
HrAverage('HrSpeed( )',6,'For HrDistance( )=1650'))
```

```
HrAverage('HrSpeed( )',6,'For '+Samd('HrDistance( )')+' and '+Same('HrCourse( )'))
```

If the above function is applied to a 1650m start, the function will return

```
HrAverage('HrSpeed( )',6,'For HrDistance( )=1650 and HrCourse( )="HV"')
```

### 16.15.16 Season( )

Returns the season number of a particular date. If no date is specified, the season number of today will be returned. If you use this function with no argument, the return value will be cached and will not be changed even if the race season changes. You should let all columns affected by the return value automatically be recalculated every time you display the view. Please do the following.

- In a starter view: Click [Modify] to open the "Edit View" form. Select the related column and click [Advanced] to show the Edit View Advanced Settings form. Then

© Welkin Systems Limited

check [No Caching].
- In a history/stat view: On the Definition of Statistical Views. Click on [View Details]. Then check [Recalc On Open].

#### Syntax

```
Season(dDate)
```

#### Example

```
Season( )
```
Returns today's season number. If today is 13 May 2011, it will return 2010 (for season 2010-2011).

```
Season({^2011/10/01})
```
Returns the season number as at date 1 Oct 2011. The return value will be 2011.

#### See Also

HrSeason( ), JrSeason( ), SrSeason( ), HrSeason2( ), JrSeason2( ), SrSeason2( ), Cache for View Data

### 16.15.17 Year( )

Year( ) is a Visual FoxPro function. It returns year number for a given date.

#### Syntax

```
Year(dExpression1)
```

#### Returns

Numeric

#### Example

```
Year({^2003/8/1})
```

Returns 2003

#### See Also

Date( ); Month( ); HrDate( ), JrDate( ), SrDate( )

## 16.16 VFP Functions & Operators

RaceMate is written in Microsoft Visual FoxPro (VFP). All rules and conventions of VFP will be applicable to RaceMate's MCL. Please visit Microsoft's web pages for Visual FoxPro to get more insight about RaceMate.

### 16.16.1 VFP Functions

See http://msdn.microsoft.com/en-US/library/f2x2b62b(v=VS.80).aspx

© Welkin Systems Limited

### 16.16.2 VFP Operators

See http://msdn.microsoft.com/en-US/library/wkz38bt3(v=VS.80).aspx

© Welkin Systems Limited

# Part XVI

# 17 How To

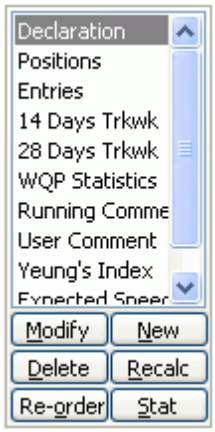
This group of unorganized topics demonstrates a set of useful features of RaceMate with examples and detailed steps.

**>>> Subtopics of this section <<<**

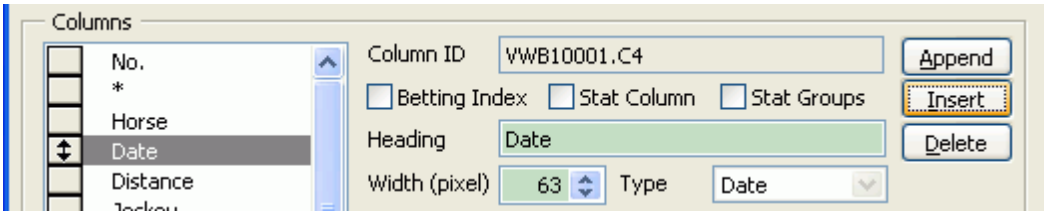
- Find horses of this race met in past starts
- Find horses which win in 60 days after a stable change
- See performance enhanced by a stable change

## 17.1 Find horses of this race met in past starts

1. Open the Declaration View.
2. Push the "Modify" button to modify the view structure.

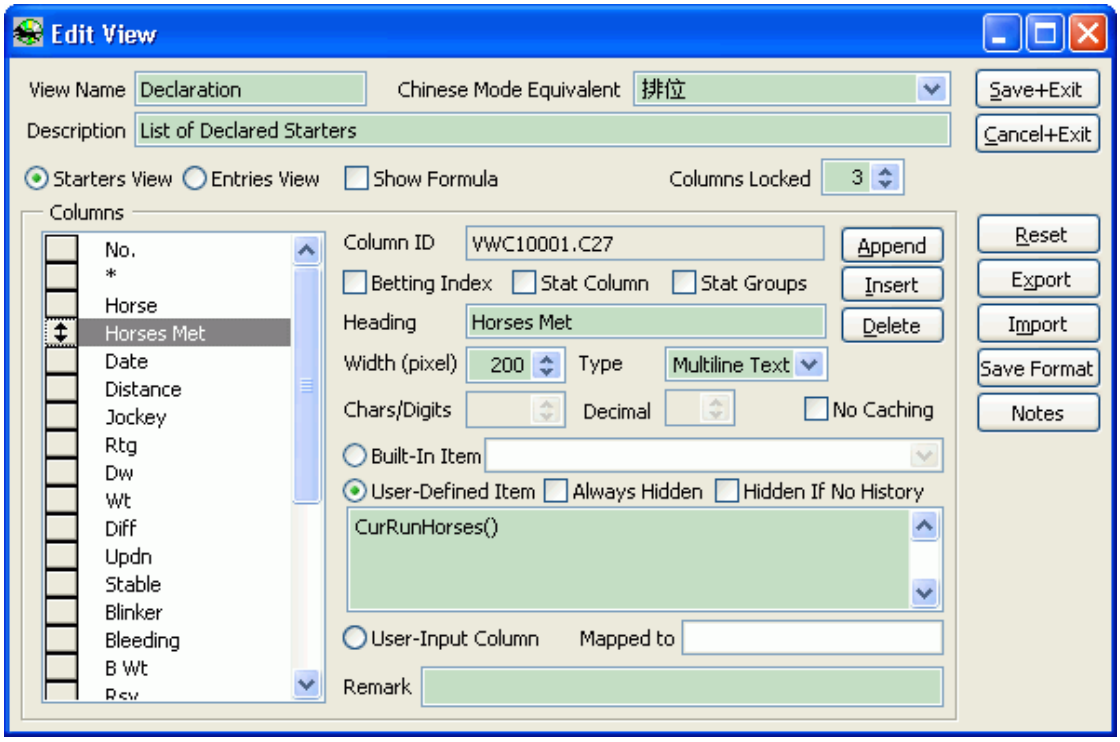



3. Highlight the column just below the "Horse" column and push the "Insert" button to add a new column.

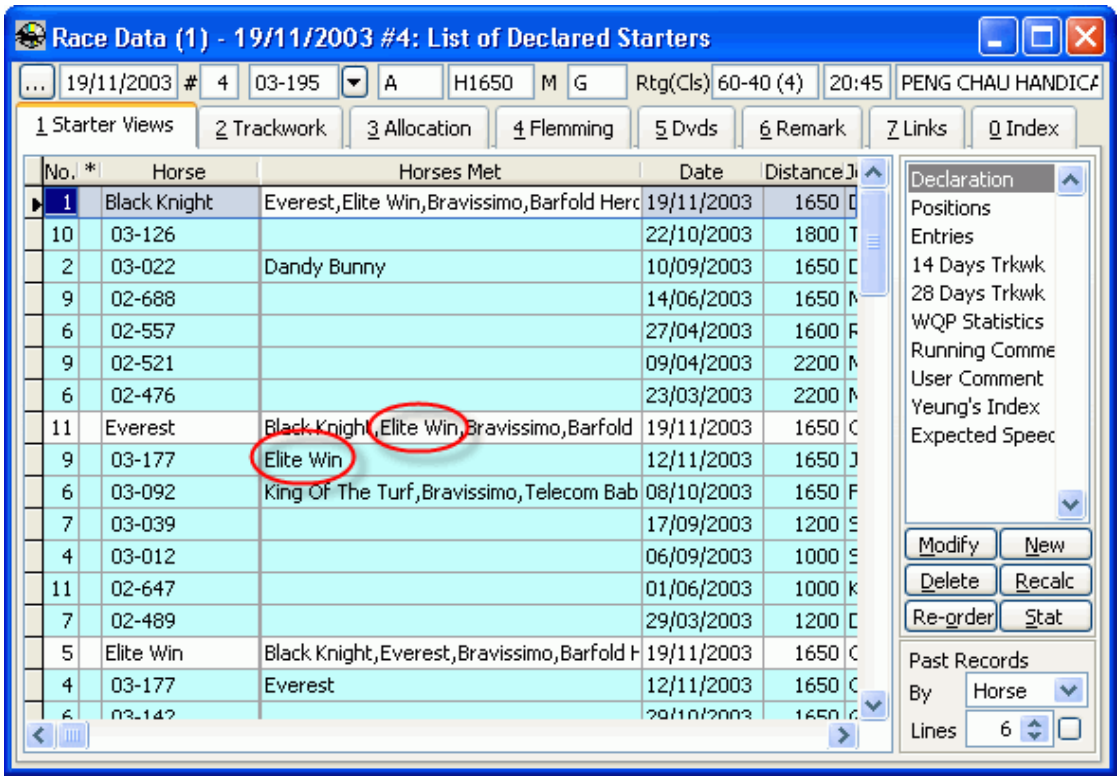



4. Name the new column "Horses Met".
5. Set it as "Multiline Text" type
6. Select User-Defined Item and enter the following formula into the formula box.

© Welkin Systems Limited



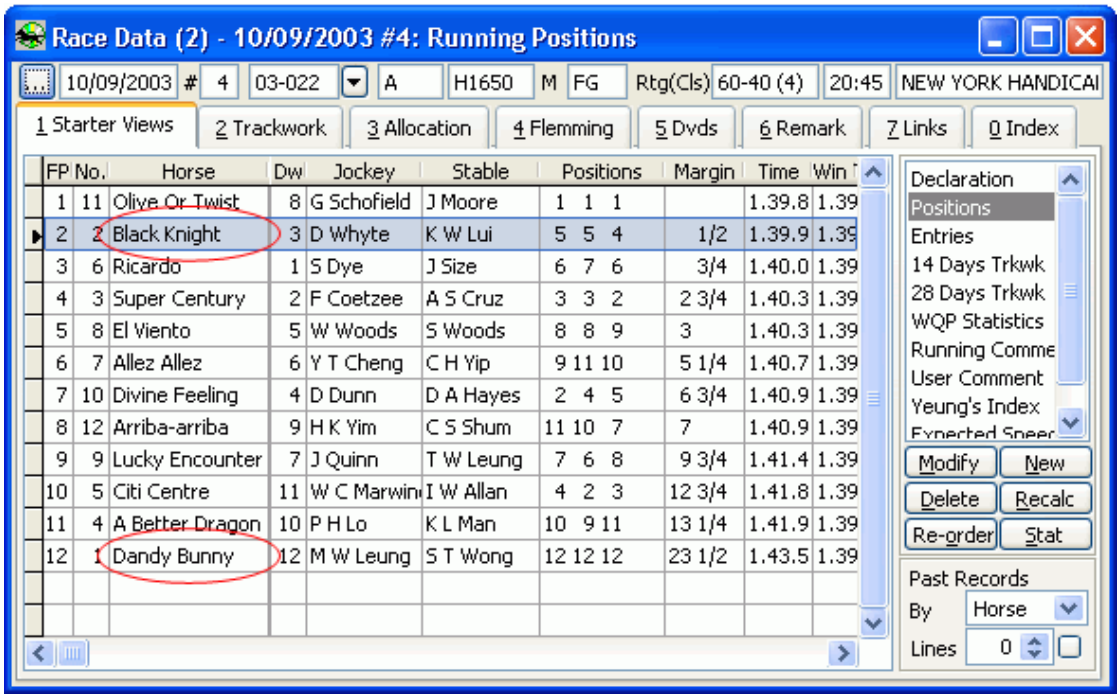

7. Push the "Save" button to save the structure and exit to the "Declaration" view.
8. Expand the history line of the view to six. You will see a view similar to the following. In the following example, you can see that Black Knight met Dandy Bunny in race 03-022 and Everest met Elite Win in race 03-177




9. If you want to get more information about the race on 10/09/2003 that Black Knight met Dandy Bunny, you may double-click on the date to open the race.

© Welkin Systems Limited

10. After the race is open, change to "Position" view to see detailed running positions and time.


Race Data (2) - 10/09/2003 #4: Running Positions
10/09/2003 # 4 03-022 A H1650 M FG Rtg(Cls) 60-40 (4) 20:45 NEW YORK HANDICAI
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
6 Remark
7 Links
0 Index
FP No. Horse Dw Jockey Stable Positions Margin Time Win
1 11 Olive Or Twist 8 G Schofield J Moore 1 1 1 1.39.8 1.39
2 2 Black Knight 3 D Whyte K W Lui 5 5 4 1/2 1.39.9 1.39
3 6 Ricardo 1 S Dye J Size 6 7 6 3/4 1.40.0 1.39
4 3 Super Century 2 F Coetzee A S Cruz 3 3 2 2 3/4 1.40.3 1.39
5 8 El Viento 5 W Woods S Woods 8 8 9 3 1.40.3 1.39
6 7 Allez Allez 6 Y T Cheng C H Yip 9 11 10 5 1/4 1.40.7 1.39
7 10 Divine Feeling 4 D Dunn D A Hayes 2 4 5 6 3/4 1.40.9 1.39
8 12 Arriba-arriba 9 H K Yim C S Shum 11 10 7 7 1.40.9 1.39
9 9 Lucky Encounter 7 J Quinn T W Leung 7 6 8 9 3/4 1.41.4 1.39
10 5 Citi Centre 11 W C Marwin I W Allan 4 2 3 12 3/4 1.41.8 1.39
11 4 A Better Dragon 10 P H Lo K L Man 10 9 11 13 1/4 1.41.9 1.39
12 1 Dandy Bunny 12 M W Leung S T Wong 12 12 12 23 1/2 1.43.5 1.39
Declaration
Positions
Entries
14 Days Trkwk
28 Days Trkwk
WQP Statistics
Running Comme
User Comment
Yeung's Index
Modify
New
Delete
Recalc
Re-order
Stat
Past Records
By Horse
Lines 0

## 17.2 Find horses which win in 60 days after a stable change

In the How-To topic See performance enhanced by a stable change we learnt how to compare the performance of a horse before and after a stable change. The following example will show you how to find all the horses which win in 60 days since a stable change. Once you understand how this example works, you will be able to create more sophisticated views, for example:

- List all the horses which have never win in its old stable, but win in 60 days after a stable change.
- List all horses which win when they runs in a new distance, which they have never tried before.

### View Structure

View Name: Win in New Stable

| Heading | MCL Function | Type (Chars) | Stat Column | Stat Group |
|---|---|---|---|---|
| Horse | (Built-In Item: Horse Name) | | | Yes |
| Old Stable Date | HrDate(1, [For Not ]+Same ('HrStableID( )')) | Date | | |
| Win in 60 Days | HHrDate( )-Column('Old Stable Date') <=60 And HrFp( )=1 | Logical | | |
| Stat Cnt | Iif(Column('Win in 60 Days'),1,0) | Numeric (1) | Yes | |

### Column: Old Stable Date

© Welkin Systems Limited

We have already explained in detail the formula of the column "Old Stable Date". If you have any problem with the column, please study the topic See performance enhanced by a stable change.

### Column: Win in 60 Days

In order to set up the criteria "win in 60 days since stable changed", we combine to logical statements with an And operator:

1. To find if a horse wins in the underlying start, we use: HrFp( )=1
2. To check if the start is in 60 days since a stable change, we use:

```
HHrDate( )-Column('Old Stable Date')<=60
```

Please note that the data type of the column "Win in 60 Days" is Logical. If both conditions are satisfied, the value of the column will be True, otherwise it will be False.

### Column: Stat Cnt

Since View Statistics can only be done on a Numeric column, therefore we add a column "Stat Cnt" which will be 1 if the column "Win in 60 Days" is true and 0 if the column "Win in 60 Days" is false. The function IIF( ) (immediate if) is used to do the job.

### What will be Shown in for a Race

If we apply the view to race number 10 on 3rd Apr 2005, we will see the data as shown in the following picture. Here are some notes about the figures in the picture:

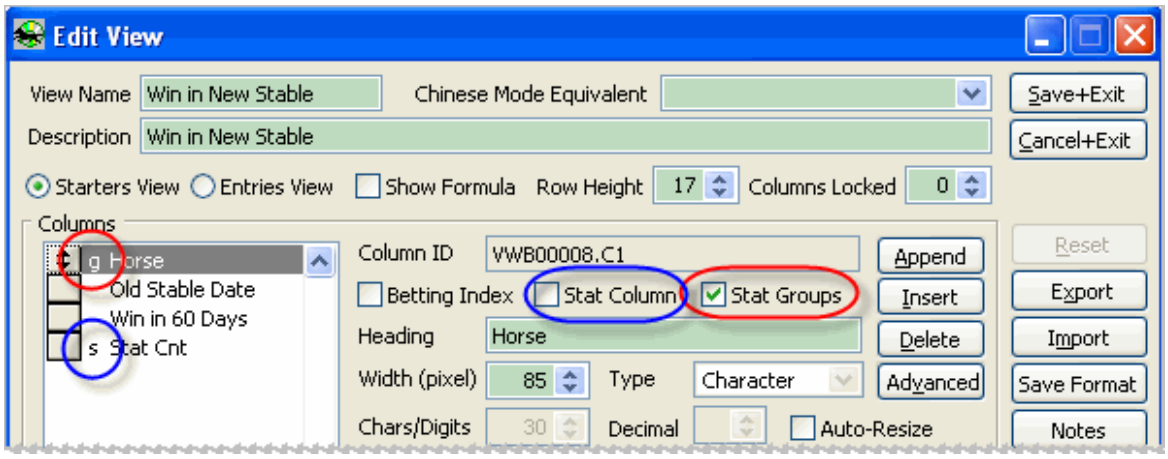
- "Very Fit" won in the underlying race. The date of the race has only been 42 days since 20 Feb 2005. Therefore the criterion "win in 60 days since stable changed" is satisfied. The column "Win in 60 Days" is thus True and the column "Stat Cnt" is 1.
- "Mighty Hugo" has no stable change and therefore the "Old Stable Date" is Null. Calculations involving a Null value will results in Null. Therefore the column "Win in 60 Days" is null. The Null value in a logical column is shown by a grey square in the checkbox. The value of the "Stat Cnt" column will be 0. Pleas note that the IIF( ) function regards a Null condition as False.
- "Royal Army" lost in the race. Therefore the column "Win in 60 Days" is false and the column "Stat Cnt" is 0.

| Horse | Old Stable Date | Win in 60 Days | Stat Cnt |
|---|---|---|---|
| Very Fit | 20/02/2005 | ☑ | 1 |
| Mighty Hugo | .NULL. | ■ | 0 |
| Leilati | .NULL. | ■ | 0 |
| Royal Army | 30/01/2005 | ☐ | 0 |
| First Among Equa | .NULL. | ■ | 0 |
| Stable Mail | .NULL. | ■ | 0 |
| Oscar Star | 26/11/2003 | ☐ | 0 |
| Flying Bishop | .NULL. | ■ | 0 |
| Delightful Win | 22/02/2004 | ☐ | 0 |
| Rice Field | .NULL. | ■ | 0 |
| Dashing Thunder | 18/01/2004 | ☐ | 0 |
| Dashing Winner | .NULL. | ■ | 0 |
| Fruity Fruity | .NULL. | ■ | 0 |
| Sparkling Star | 20/06/2004 | ☐ | 0 |

© Welkin Systems Limited

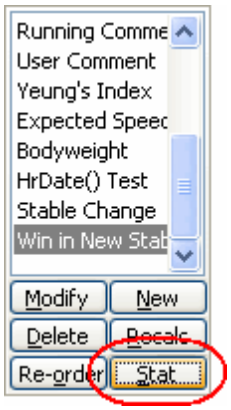
### Stat Column and Stat Groups

In order to do a View Statistics, a Stat Column must be found in the view. A Stat Column must be of a Numeric data type. RaceMate will sum up the value of the Stat Column as the statistical result. We may carry out statistics in groups. In this sample view, the column Horse is set as a Stat Group column and a horse will be a group.



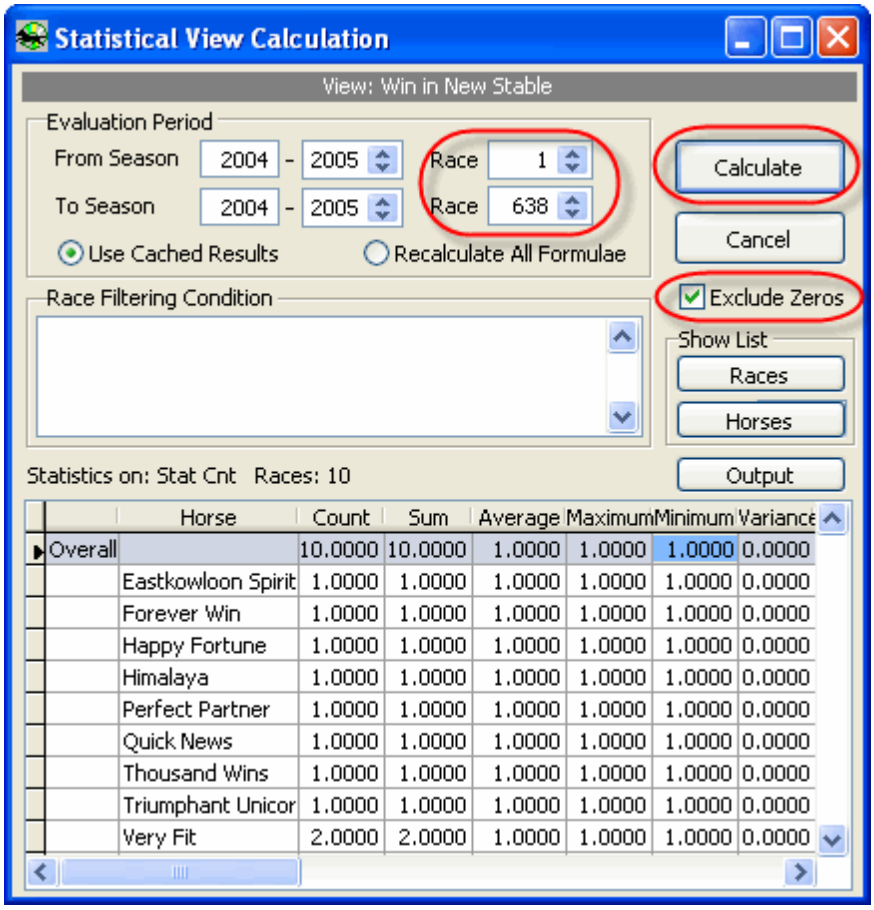

### Do the Statistics

To do the statistics, click on the [Stat] button at the right hand side.



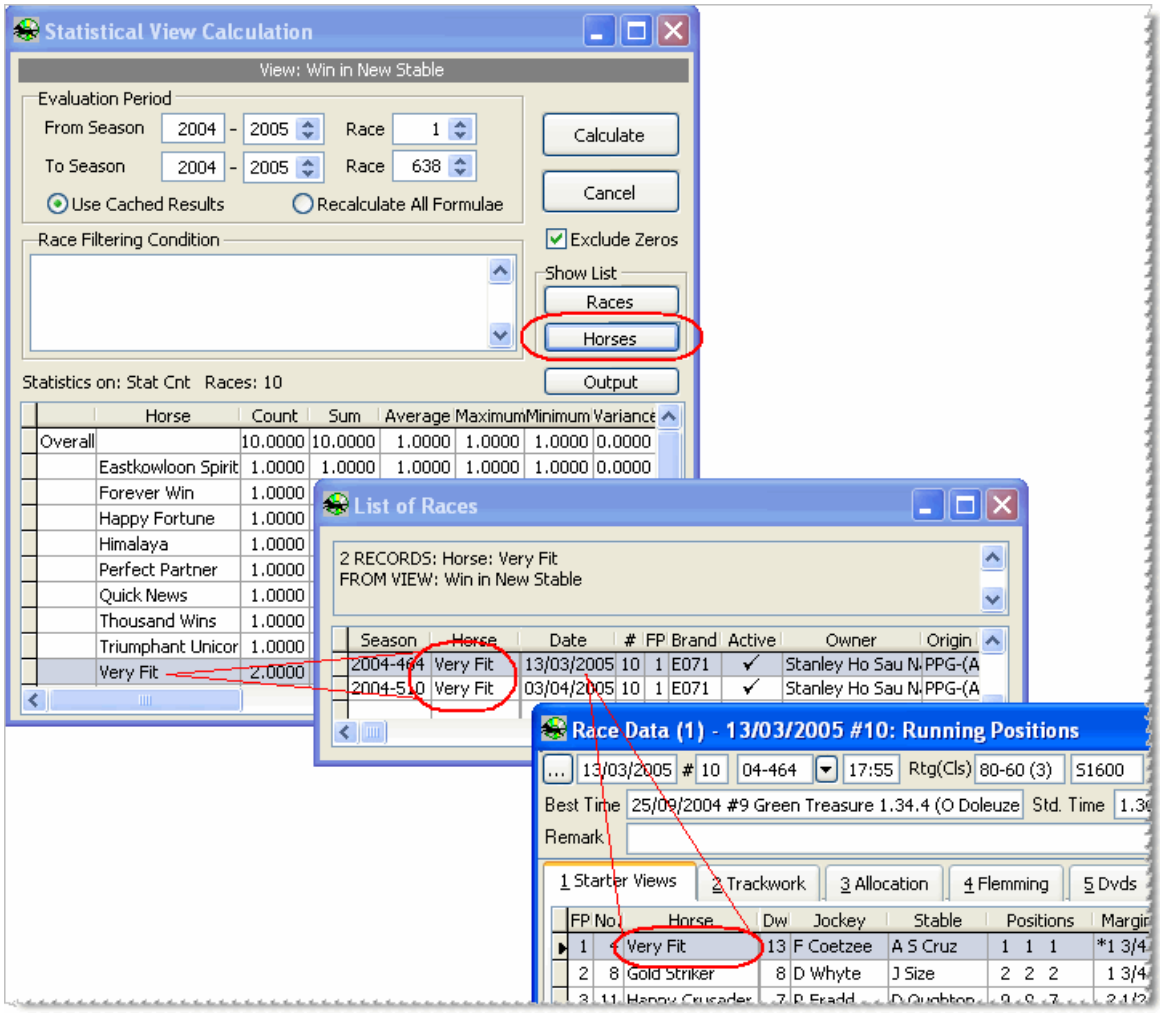

The following form will appear. Select [Exclude Zero] (since we are interested in only the horses with a 1 in the "Stat Cnt" column. The following figure shows the statistical result for the period races 1-638 of the season 2004-2005. It shows that "Very Fit" has won twice since a stable change.

© Welkin Systems Limited


Statistical View Calculation
View: Win in New Stable
Evaluation Period
From Season 2004 - 2005
To Season 2004 - 2005
Race 1
Race 638
Use Cached Results
Recalculate All Formulae
Calculate
Cancel
Exclude Zeros
Race Filtering Condition
Show List
Races
Horses
Output
Statistics on: Stat Cnt Races: 10
Horse Count Sum Average Maximum Minimum Variance
Overall 10.0000 10.0000 1.0000 1.0000 1.0000 0.0000
Eastkowloon Spirit 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Forever Win 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Happy Fortune 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Himalaya 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Perfect Partner 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Quick News 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Thousand Wins 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Triumphant Unicor 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Very Fit 2.0000 2.0000 1.0000 1.0000 1.0000 0.0000

If you want to show the starts of "Very Fit", double click on the name "Very Fit". The starts will be shown in a separate form. You may hold down the Ctrl key and double-click on a race date on the form to show the details of the Race.

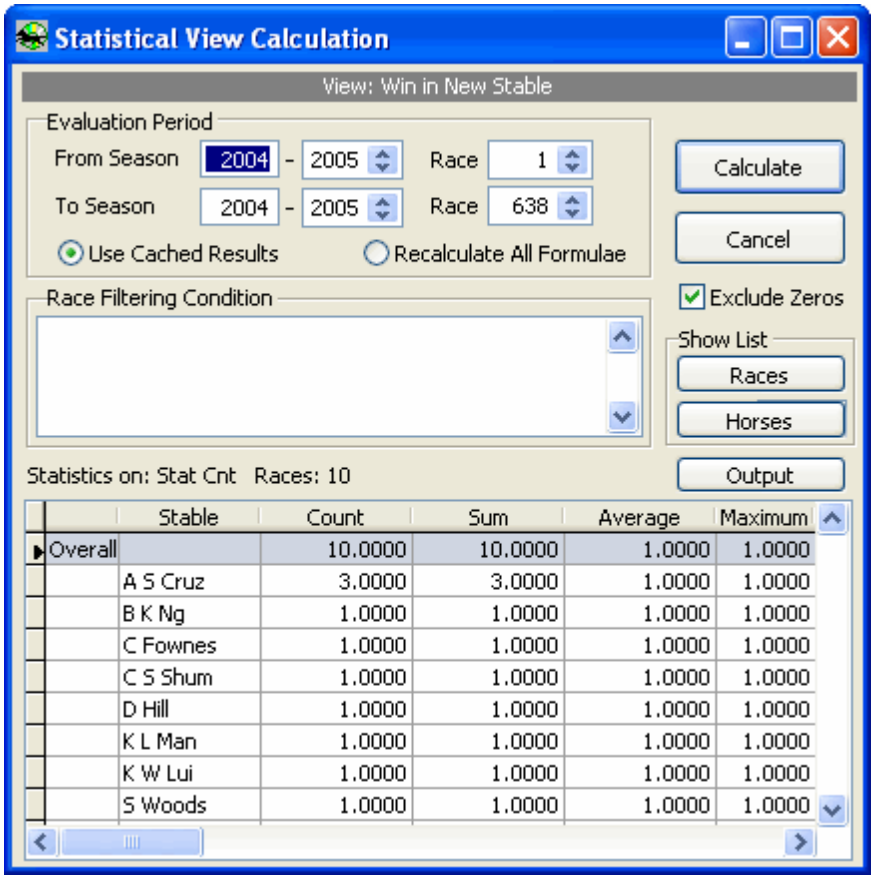
© Welkin Systems Limited


Statistical View Calculation
View: Win in New Stable
Evaluation Period
From Season 2004 - 2005 Race 1
To Season 2004 - 2005 Race 638
Use Cached Results
Recalculate All Formulae
Race Filtering Condition
Calculate
Cancel
Exclude Zeros
Show List
Races
Horses
Output
Statistics on: Stat Cnt Races: 10
Horse Count Sum Average Maximum Minimum Variance
Overall 10.0000 10.0000 1.0000 1.0000 1.0000 0.0000
Eastkowloon Spirit 1.0000 1.0000 1.0000 1.0000 1.0000 0.0000
Forever Win 1.0000
Happy Fortune 1.0000
Himalaya 1.0000
Perfect Partner 1.0000
Quick News 1.0000
Thousand Wins 1.0000
Triumphant Unicor 1.0000
Very Fit 2.0000
List of Races
2 RECORDS: Horse: Very Fit
FROM VIEW: Win in New Stable
Season Horse Date # FP Brand Active Owner Origin
2004-464 Very Fit 13/03/2005 10 1 E071 Stanley Ho Sau N. PPG-(A
2004-510 Very Fit 03/04/2005 10 1 E071 Stanley Ho Sau N. PPG-(A
Race Data (1) - 13/03/2005 #10: Running Positions
13/03/2005 # 10 04-464 17:55 Rtg(Cls) 80-60 (3) S1600
Best Time 25/09/2004 #9 Green Treasure 1.34.4 (O Doleuze Std. Time
Remark
1 Starter Views
2 Trackwork
3 Allocation
4 Flemming
5 Dvds
FP No. Horse Dw Jockey Stable Positions
1 4 Very Fit 13 F Coetzee A S Cruz 1 1 1 *1 3/4
2 8 Gold Striker 8 D Whyte J Size 2 2 2 1 3/4

### Group by Stable

If we change the horse name column to stable name (by selecting the Stable Name built-in item) and re-do the statistics we can compare the figures between difference stables:

© Welkin Systems Limited


Statistical View Calculation
View: Win in New Stable
Evaluation Period
From Season 2004 - 2005 Race 1
To Season 2004 - 2005 Race 638
Use Cached Results
Recalculate All Formulae
Calculate
Cancel
Exclude Zeros
Race Filtering Condition
Show List
Races
Horses
Statistics on: Stat Cnt Races: 10
Output
Stable Count Sum Average Maximum
Overall 10.0000 10.0000 1.0000 1.0000
A S Cruz 3.0000 3.0000 1.0000 1.0000
B K Ng 1.0000 1.0000 1.0000 1.0000
C Fownes 1.0000 1.0000 1.0000 1.0000
C S Shum 1.0000 1.0000 1.0000 1.0000
D Hill 1.0000 1.0000 1.0000 1.0000
K L Man 1.0000 1.0000 1.0000 1.0000
K W Lui 1.0000 1.0000 1.0000 1.0000
S Woods 1.0000 1.0000 1.0000 1.0000

## 17.3 See performance enhanced by a stable change

The following table shows the structure of a view for displaying:

1. the date on which the stable of a horse changes
2. the WQP statistics of a horse before stable change
3. the WQP statistics of a horse after stable change

**View: Stable Change**

| Heading | User-Defined MCL Function | Type (Chars) |
|---|---|---|
| Horse | (Built-In Item: Horse Name) | |
| Old Stable Date | `HrDate(1, [For Not ]+Same('HrStableID( )'))` | Date |
| Before Stable Change | `HrWQPStat(,'For HrDate( )<='+Qstr(Column('Old` | Character (23) |
| After Stable Change | `HrWQPStat(,'While HrDate( )>'+Qstr(Column('Old` | Character (23) |

Apply the view to the 6th race on 1 May 2005, the following result will be shown. The statistical figures show that, in this race, there are performance gains after the horses were moved into new stables.

© Welkin Systems Limited

| Horse | Old Stable Date | Before Stable Change | After Stable Change |
|---|---|---|---|
| Easy Game | 20/06/2004 | 0, 0, 0, 9 ( 9) | 2, 2, 0, 6 ( 10) |
| Fortune Warrior | .NULL. | .NULL. | .NULL. |
| Diamond Era | .NULL. | .NULL. | .NULL. |
| Mr Beautiful | 18/06/2003 | 1, 0, 0,14 ( 15) | 1, 1, 4, 9 ( 15) |
| Speed Of Light | .NULL. | .NULL. | .NULL. |
| Great Star | .NULL. | .NULL. | .NULL. |
| Main Attraction | 11/05/2002 | 0, 0, 1, 2 ( 3) | 2, 2, 0,17 ( 21) |
| The Lavish Lad | .NULL. | .NULL. | .NULL. |
| Bullish General | .NULL. | .NULL. | .NULL. |
| Maximus | .NULL. | .NULL. | .NULL. |
| Right Choice | .NULL. | .NULL. | .NULL. |
| Liverbird | .NULL. | .NULL. | .NULL. |
| Harry Harry | 17/10/2004 | 0, 0, 1, 8 ( 9) | 1, 0, 1, 3 ( 5) |
| Just Too Good | .NULL. | .NULL. | .NULL. |

The following paragraphs will explain what are shown in the view, how this view is constructed, and some useful technical tips. The content is deliberately structured so that it will be understood by users with little MCL experience. Even if you are an intermediate to advanced level MCL user, you will find some of the information useful. If you have no knowledge or experience on how to create a starter views, you may find the topic Creating a Starter View helpful.

### Find the Date of a Particular Start

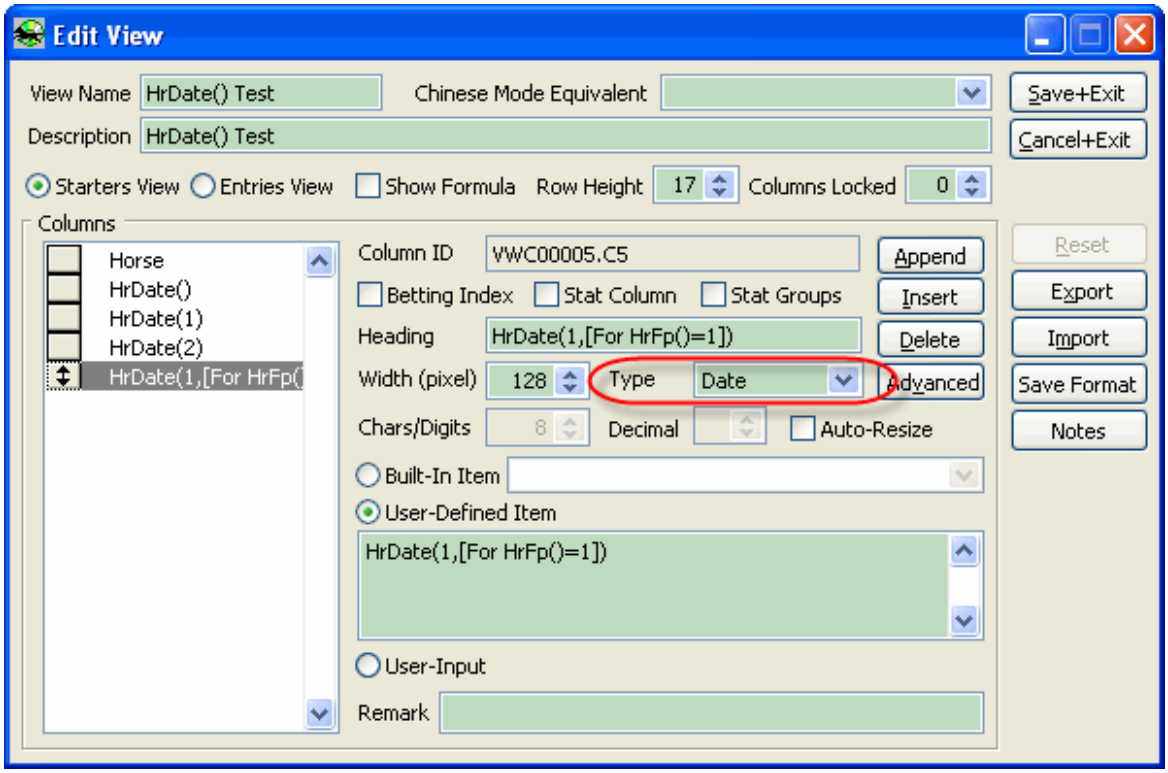
To find the date of a particular start, we use the HrDate( ) function. When no argument is passed, the function will return the date of the viewing race. Passing an argument 1 into the function, i.e., HrDate(1), it will return the date of the horses' last start. Passing 2, i.e., HrDate(2), the function will return the date of the start prior to the last start, and so on. We can also specify a filter condition by passing an expression to the 2nd argument. For example, the following function will return the date of the last winning start of a horse:

```
HrDate(1, [For HrFp( )=1])
```

The first argument of the function above indicates we want the last start. However it is not an ordinary "last", but the last start satisfying the condition specified by the 2nd argument "For HrFp( )=1". Since HrFp( ) returns the finishing position of the horse, thus, the date of the last winning start will be returned.

We may put the above functions into a view to see the results. Please make sure "Date" type is selected for the columns as show in the following figure:

© Welkin Systems Limited




Apply the view to the 6th race of the meeting on 1 May 2005. We will see the following result:

| Horse | HrDate() | HrDate(1) | HrDate(2) | HrDate(1,[For HrFp()=1]) |
|---|---|---|---|---|
| Easy Game | 01/05/2005 | 13/04/2005 | 23/03/2005 | 13/04/2005 |
| Fortune Warrior | 01/05/2005 | 09/04/2005 | 05/03/2005 | .NULL. |
| Diamond Era | 01/05/2005 | 26/03/2005 | 19/03/2005 | .NULL. |
| Mr Beautiful | 01/05/2005 | 13/04/2005 | 23/03/2005 | 04/11/2003 |
| Speed Of Light | 01/05/2005 | 03/04/2005 | 05/02/2005 | 20/06/2004 |
| Great Star | 01/05/2005 | 13/04/2005 | 19/03/2005 | 16/01/2005 |
| Main Attraction | 01/05/2005 | 09/04/2005 | 19/03/2005 | 01/05/2003 |
| The Lavish Lad | 01/05/2005 | 03/04/2005 | 26/03/2005 | .NULL. |
| Bullish General | 01/05/2005 | 26/03/2005 | 05/03/2005 | .NULL. |
| Maximus | 01/05/2005 | .NULL. | .NULL. | .NULL. |
| Right Choice | 01/05/2005 | 13/04/2005 | 09/03/2005 | .NULL. |
| Liverbird | 01/05/2005 | 13/04/2005 | 09/04/2005 | .NULL. |
| Harry Harry | 01/05/2005 | 03/04/2005 | 26/03/2005 | 26/03/2005 |
| Just Too Good | 01/05/2005 | 13/04/2005 | 23/03/2005 | 27/03/2004 |

In the figure above, we see that the latest start (relative to the viewing race on 1/5) of the horse Easy Game was on 13 Apr 2005 and it was the winning horse of that race.

### Date on Which a Horse was in Its Old Stable

To show the trainer of a horse, we use the MCL function HrStable( ). Suppose the trainer of a horse in the viewing race is "A S Cruz". We can use the following function to find when the horse was still trained by its prior trainer:

```
HrDate(1, [For Not HrStable( )='A S Cruz'])
```

© Welkin Systems Limited

However, since almost all horses are in different stables, we may not hard-code the stable name into our functions as shown in the above line. To solve this problem, we may use the magic function Same( ) and rewrite the above line as:

```
HrDate(1, [For Not ]+Same('HrStable( )'))
```

Same('HrStable( )') is actually an abbreviation of [HrStable( )=]+Qstr(HrStable( )). When the trainer of a horse in the viewing race is A S Cruz, Same('HrStable( )') will be expanded to [HrStable( )='A S Cruz'] before processing. Similarly, if the trainer of the horse is T W Leung, Same('HrStable( )') will be expanded to [HrStable( )='T W Leung'].

### WPQ Statistics

The MCL function HrWQPStat( ) gives us the WQP statistics. We may specify a filter condition in the 2nd argument. For example, to find the WQP statistics for turf tracks, we write:

```
HrWQPStat( ,[For HrIsTurf( )])
```

To find the WQP statistics in other tracks, we write:

```
HrWQPStat( ,[For Not HrIsTurf( )])
```

In order to compare the performance of the horses before and after a particular date, say 1/1/2005, we may use the following functions:

```
HrWQPStat( ,[For HrDate( )>={^2005/1/1}])        && WQP stat after 1/1/2005
HrWQPStat( ,[For Not HrDate( )<{^2005/1/1}])     && WQP stat before 1/1/2005
```

There may be some horses that arrived after 1/1/2005. The statistical figures returned by the functions will all be zero. In order to clearly show the difference between no winning (zero) and condition not applicable, we can give a default value .NULL. to the 3rd argument of the functions making the function return .NULL. for the horses for which the condition is not applicable, as shown in the following examples:

```
HrWQPStat( ,[For HrDate( )>={^2005/1/1}],.Null.)      && WQP stat after 1/1/2005
HrWQPStat( ,[For Not HrDate( )<{^2005/1/1}],.Null.)   && WQP stat before 1/1/2005
```

### Comparing Performance Difference Before and After a Stable Change

Putting the above basic knowledge all together, it will not be difficult for us to create a view to show the performance difference before and after a stable change. The structure of the view is listed as follows:

| Heading | User-Defined MCL Function | Type (Chars) |
|---|---|---|
| Horse | (Built-In Item) | |
| Old Stable Date | `HrDate(1, [For Not ]+Same('HrStable( )'))` | Date |
| Before Stable Change | `HrWQPStat(,'For HrDate( )<='+Qstr(Column('Old` | Character (23) |
| After Stable Change | `HrWQPStat(,'For HrDate( )>'+Qstr(Column('Old S` | Character (23) |

### Using "While" to Make a View Faster

© Welkin Systems Limited

In order to make the view show up faster, we can replace the keyword "For" with "While" in the condition expression of the "After Stable Change" column (but not the "Before Stable Change" column). The difference between the keywords is:

- Using For, RaceMate will scan all past starts to find the ones which satisfy the condition.
- Using While, RaceMate will start from the latest start and scan the past starts one by one. Once a start which does not satisfy the condition is found, the scanning action will stop immediately and the older starts won't be touched anymore.

When we are doing the "After Stable Change" statistics, the races belong to the old stable is not application. Therefore using while will significantly increase the statistical speed.

### HrStableID( ) Against HrStable( )

The HrStableID( ) function returns a 2-char ID of a stable, which is faster, language independent, and more accurate than using the full name. When we do statistics, comparing an ID is always preferable to comparing a name. Therefore another enhancement to the view is to change the HrStable( ) function to HrStable( ).

After making the above improvements to the view, the view becomes:

| Heading | User-Defined MCL Function | Type (Chars) |
|---|---|---|
| Horse | (Built-In Item) | |
| Old Stable Date | `HrDate(1, [For Not ]+Same('HrStableID( )'))` | Date |
| Before Stable Change | `HrWQPStat(,'For HrDate( )<='+Qstr(Column('Old Stab` | Character (23) |
| After Stable Change | `HrWQPStat(,'While HrDate( )>'+Qstr(Column('Old Sta` | Character (23) |

### Enhancing the View According to Your Need

You may add more filtering conditions and columns to the view according to your needs. For example, if you are only interested in the horses which change stable during the past 12 months, your may change of condition of the "Old Stable Date" column to:

```
HrDate(1, [While HrDaysAgo( )<=365 For Not ]+Same('HrStable( )'))
```

© Welkin Systems Limited

# Part XVI

# 18 Questions & Answers

This section provides help on common problems that RaceMate users may encounter.

If you still cannot find solutions in this user guide or the user manuals, or you find any mistakes in our data or programs, please contact RaceMate's Technical Support Team at:

E-mail : Contact@RaceMateNet.Com
Phone : (852) 2801-7754
Fax : (852) 2521-4642

**>>> Subtopics of this section <<<**


- Questions You May Asked After Upgrading to 9.0
- How Can I Display Chinese in English Windows
- Chinese Characters Becomes Garbage
- RaceMate Frozen When Starter View Being Opened


## 18.1 Questions You May Asked After Upgrading to 9.0

1. Where is the "History" page on the Horse, Jockey, and Stable data forms?

The "History" page on the forms has been replaced by a new "Stat Views" page. If you double-click on the [*Basic Columns] view on the view list of the page, you will see a view similar to the History page of RaceMate 8.0.

2. Can I modify the [*Basic Columns] view, such as moving columns, to satisfy my need?

You can modify the [*Basic Columns] view. However RaceMate will reset all system built-in views, including [*Basic Columns], when it restarts and you change will be lost. In order to keep your change, you may save the [*Basic Columns] view as a new view. On this new view, you may then add or remove columns. Your changes will be kept forever.

3. After I save the [*Basic Columns] view as a new view, the original [*Basic Columns] view will no longer be needed. Is it possible to delete it?

A built-in view can't be deleted. However you may hide it by pushing the [View Details] button on the "Definition of Statistical Views" form and unchecking corresponding checkboxes at the bottom.

4. On a new Stat View I am not able to drag a column left or right to a new position. Why?

On a Stat View columns are not allowed to be dragged to another position directly. You need to click on the [Edit View Structure] button to open the "Definition of Statistical Views" form and click on the [Up] and [Down] buttons at the right bottom to change the position of a column.

5. On a Stat View, I select "1 Year" as data range. However I find that there are

© Welkin Systems Limited

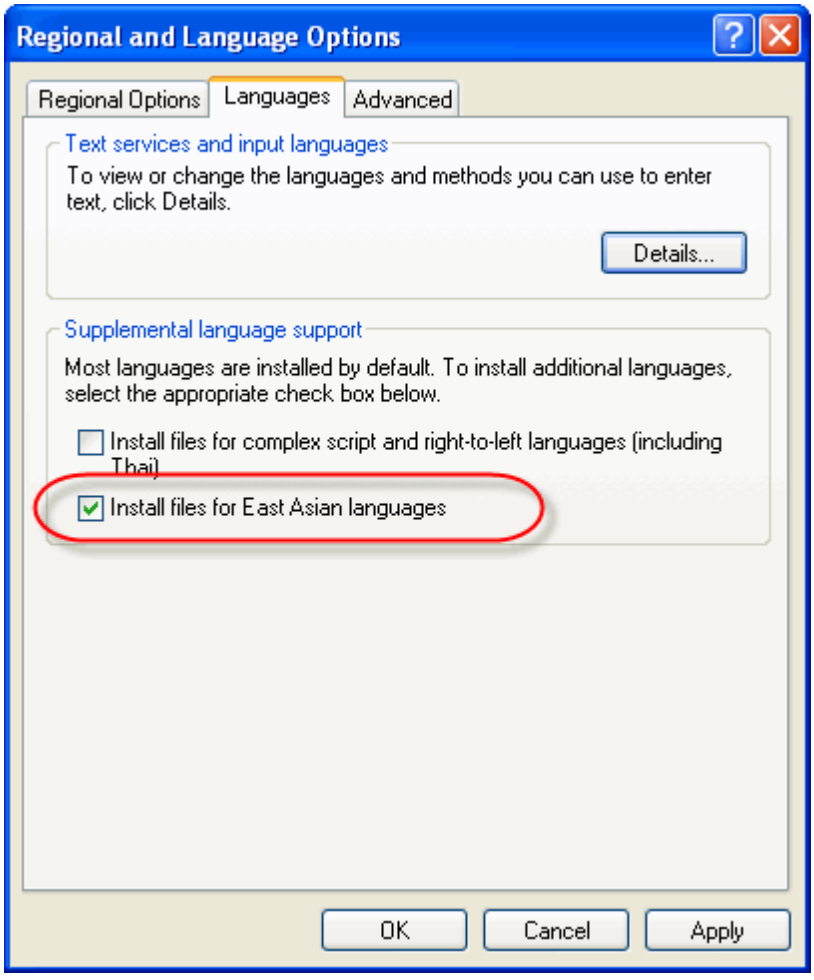
races is earlier than 1 year ago retrieved. Why?

Year (and other range type) is counted from the latest race rather than today. Therefore if the latest race took place long ago, the earliest race retrieved could be far beyond 1 year ago.

## 18.2 How Can I Display Chinese in English Windows

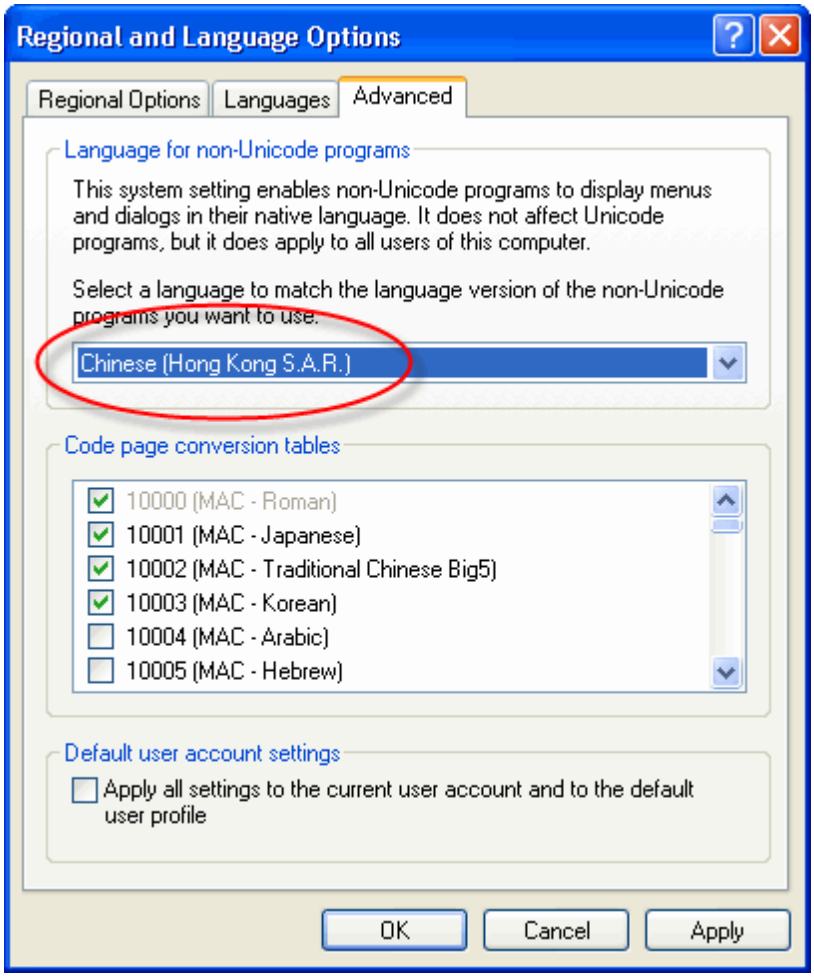
Follow the steps below to enable Chinese display in a pure English Windows:

1. Open Windows Control Panel and double-click on the Regional and Language Options icon (in the Date, Time, Language, and Regional Options group).
2. After the Regional and Language Options form is shown, switch to the "Languages" page
3. Check the [Install files for East Asian languages box


Regional and Language Options
Regional Options
Languages
Advanced
Text services and input languages
To view or change the languages and methods you can use to enter text, click Details.
Details...
Supplemental language support
Most languages are installed by default. To install additional languages, select the appropriate check box below.
Install files for complex script and right-to-left languages (including Thai)
Install files for East Asian languages
OK
Cancel
Apply

4. Click on the [Apply] button. Windows may ask you to insert your Windows disk to continue
5. On the "Advanced" page of the "Regional and Language Options" form, choose [Chinese (Hong Kong S.A.R.)] on the dropdown list.

© Welkin Systems Limited


Regional and Language Options
Regional Options
Languages
Advanced
Language for non-Unicode programs
This system setting enables non-Unicode programs to display menus and dialogs in their native language. It does not affect Unicode programs, but it does apply to all users of this computer.
Select a language to match the language version of the non-Unicode programs you want to use.
Chinese (Hong Kong S.A.R.)
Code page conversion tables
10000 (MAC - Roman)
10001 (MAC - Japanese)
10002 (MAC - Traditional Chinese Big5)
10003 (MAC - Korean)
10004 (MAC - Arabic)
10005 (MAC - Hebrew)
Default user account settings
Apply all settings to the current user account and to the default user profile
OK
Cancel
Apply

6. After rebooting Windows, start RaceMate and choose [Change to Chinese Display] on the [Tools] menu of the RaceMate.

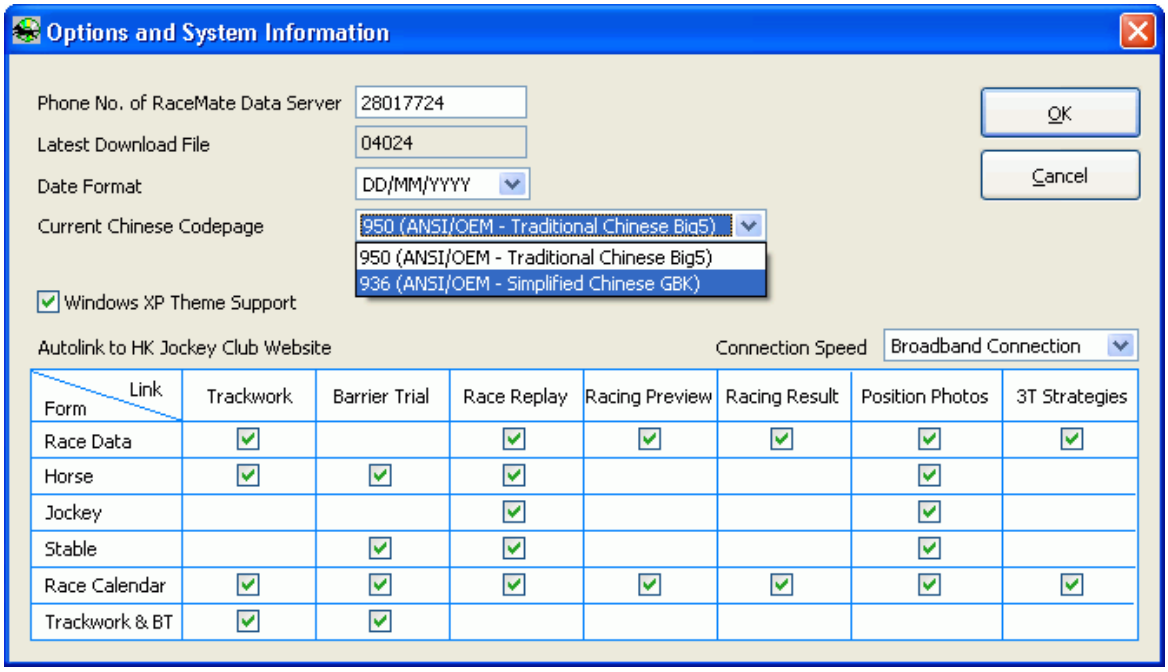
## 18.3 Chinese Characters Becomes Garbage

No matter you are using a Tradition Chinese Windows or a Simplified Chinese Windows, RaceMate will correctly detect the codepage of your Windows and properly display Chinese characters. However if you are using an external Chinese system, such RichWin and China Star, on top of an English Windows, you have to tell RaceMate which codepage you are using, otherwise Chinese characters may become garbage. Follow the steps below to choose a correct codepage:

**RaceMate Edition**

1. Press Ctrl+L to change RaceMate to English Mode.
2. Select [Options and System Information] from the [Tools] menu to show the Options and System Information form as shown below. Choose a correct codepage from the [Current Chinese Codepage] dropdown.

© Welkin Systems Limited




**RaceMate Standard Edition**

Since there is no English Mode in RaceMate Standard Edition, there may be problems opening and viewing the content of the Options and System Information Form if all Chinese characters become garbage. Alternatively, you may open the text file RMATE.INF which resides in the home folder of RaceMate. (If you can't find the file, you may create one yourself.) Add the following section in the file:

[International]
ChineseCodePage=936 (ANSI/OEM - Simplified Chinese GBK)

The sample above assumes you are using a Simplified Chinese Windows. If you are using a Traditional Chinese Windows, the lines should be:
[International]
ChineseCodePage=950 (ANSI/OEM - Traditional Chinese Big5)

## 18.4 RaceMate Frozen When Starter View Being Opened

While you are editing a starter view, if you have input a function or any strange things which have a conflict with the system, you may no longer be able to open this starter view. When you try to open the view, RaceMate freezes up. Since you are not able to open the view, you will not be able to modify and correct what you input. Although such problem is extremely rare, in case you encounter such a problem, you may follow the steps below to fix it:

**Suppress Calculation When a View is Opened**

- If you can open the Race Data form, but RaceMate freezes up when you try to switch to a view, you can simply hold down both the [Ctrl] and [Shift] key before you select the target view. Keep the keys down until the target view is successfully opened.
- If you can't open the Race Data form at all, you may hold down both the [Ctrl] and [Shift] key before you try to open the Race Data form. Keep the keys down until the Race Data form is successfully opened.

© Welkin Systems Limited

RaceMate will not do any data retrieval or calculation for a view opened in the above ways. You will see a blank view without any horse data. After the view is successfully opened, you may then modify the formulae of the view to fix the problem.

© Welkin Systems Limited

# Part

# 19 Technical Reference

We will discuss a couple of technical reference data in this chapter.

- Data Range for Hong Kong Edition

## 19.1 Data Range for Hong Kong Edition

The Hong Kong Jockey Club adds and removes data items from time to time. When you are going to carry out statistics, you must know the following data range in tables.

| Data Item | Jockey Club Added or RaceMate Added / Changed | Remarks and Related MCL functions (but not all) |
|---|---|---|
| Barrier Trials | 27/08/1991 | HrActivity( ), HrTwCount( ) |
| Gears | 09/1991 | HrBliner( ), HrBlinkerAbbr( ), HrBlinerAbbrNew( ), HrBlinkerNew( ) |
| Race Type | 09/1994 | IrregularRace( ) |
| Apprentice Allowance | 8/11/1997 | HrWeightx( ), HrAllowance( ) |
| Rating Range of a Race | 09/2000 | HrRaceRating( ) |
| Overseas Record | 24/03/2002 | |
| Horse Section Time | 12/05/2008 | HrSectTime( ), HrSectTimew( ) |
| 2 Decimal Places Time | 12/05/2008 | HrTimew( ), HrSeconds( ) |
| Horse Section Positions | 12/05/2008 | HrSectPositions( ) |
| Horse Section Margins | 12/05/2008 | HrSectMargins( ), HrSectMarginsx ( ), HrSectMargLens( ), HrSectMargLensx( ) |
| Prize for Races | 09/2008 | HrRacePrize( ) |
| Horse Stakes | 09/2008 | HrStakes( ) |
| Track Work Time with 2 Decimal Places | 09/09/2008 | |
| Penetrometer and Clegg Hammer Readings | 15/9/2008 | HrPenetrometer( ) |
| Veterinary Record | 08/11/2008 | RaceMate reads veterinary records of all horses which are active on or after 08/11/2008, before which the records are maintained manually. HrSickHistory( ), HrBleedDate( ), HrBleedDateStr( ) |
| Special Bit Report | 17/12/2008 | HrSpecialBit( ), HrSpecialBitAbbr( ) |
| Time to Collect Odds | 7/3/2009 | During a long period of time before 6/3/2009, the collection time of the 3 odds is: 12:00 am, 5 minutes before race, and the time the race starts. Since 7/3/2009, it has be changed to 8:00 pm on prior day, 1 hour before race, and the time the race starts. |
| Jockeys' Clothes | 19/1/2010 | Clothes of retired horses may be missing. |
| Speed Power Data | 1/9/2010 | HrPowerFigure( ), |

© Welkin Systems Limited

| | | |
|---|---|---|
| | | HrPowerComments( ) |
| Overweight | 1/9/2011 | HrOverweight( ) |

Remarks: In order to speed up the download speed of the setup program, some time ago, we had cut data before 1997 in the SETUP-HK.EXE of the Hong Kong Edition. This will not be a problem for most users. However, if you find that this range of data is important for you and the data for season 1987-1996 are really missing from your RaceMate, you may click on the following links to download the latest SETUP-HK.EXE program and run it again. Running the SETUP-HK.EXE program once again will not erase your personal data, including notes and views. You will not be required to register your software again.

Link to download SETUP-HK.EXE: (Download only if 86-97 data is necessary for you. Before downloading, please verify the data is really missing from your RaceMate.)
http://www.racematenet.com/setup/hk/setup-hk.exe

© Welkin Systems Limited

# Part

# 20 License Agreement

IMPORTANT - READ CAREFULLY: This License Agreement is a legal agreement for RaceMate between you (either an individual or a single entity) and Welkin Systems Limited (abbreviated as Welkin). By installing RaceMate, you agree to be bound by the terms of this agreement.

**DEFINITIONS**

- Welkin means Welkin Systems Limited and its suppliers and licensors, if any.
- Information Technology Research Centre is a subsidiary of Welkin Systems Limited. Information Technology Research Centre Develops RaceMate and Welkin owns the copyright of RaceMate and markets the product.
- RaceMate means any version of RaceMate software. The term RaceMate also includes associated media, printed materials, and online and electronic documentations.

**GRANT OF LICENSE**

This license agreement grants you the following rights:

- Installing and Running RaceMate: You may install RaceMate in more than one computer. However, at any time, you may activate, use, access, run, or otherwise interact with only one copy of RaceMate.
- Reservation of Rights: All rights not expressly granted are reserved by Welkin.
- Not for Resale: You cannot resale your RaceMate to any other party.
- Reverse Engineering, Decompilation, and Disassembly: You may not reverse engineer, decompile, or disassemble RaceMate.
- Separation of Components: RaceMate is licensed as a single product. Its component parts may not be separated for use on more than one computer.
- Rental: You may not rent, lease, or lend RaceMate.
- Copying and Backing up: You may copy or back up programs or data of RaceMate. However, you are not allowed to copy RaceMate programs or data (even part of it) to any other party.

**UPGRADES**

If this copy of RaceMate is an upgrade from an earlier version of RaceMate, it is provided to you on a license exchange basis. You agree by your installation and use of this copy of the RaceMate to voluntarily terminate your earlier license agreement.

**COPYRIGHT**

All data, programs and designs of RaceMate (including but not limited to any data, electronic documents, images, photographs, text, and utilities incorporated into the RaceMate) are owned by Welkin and corresponding suppliers, if any, and are protected by applicable copyright and intellectual property laws and treaties.

**PROGRAM UPDATE**

While you download and update RaceMate's data from Welkin's data server, Welkin may change designs, enhance or drop functions and change data and data structure of RaceMate without prior notice.

**DISCLAIMER**

*© Welkin Systems Limited*

Welkin disclaims any responsibility and accepts no liability for any direct or indirect loss or damage arising from any inaccuracies, omissions or typographical errors of RaceMate and its data and documentations. Welkin also does not warrant the accuracy, completeness, timeliness or fitness for purpose of RaceMate data. Welkin shall not be liable for any failure to supply data due to telephone line or Internet failure, power failure or interruption, Acts of God, controls or restrictions of the Jockey Club or government, cessation or failure of data supply by the Jockey Club, or any other causes beyond Welkin's control.

© Welkin Systems Limited

# Index

## - A -

About RaceMate 247
About Upgrading Windows 32
Absence of Values 257
Accuracy of Prediction Model 218
Advanced Settings of View Columns 135
All Distances WQP Stats View 168
Allocation Page (Race Data Form) 98
Argument of History/Stat View 149, 162
Arguments of MCL functions 253
Arithmetic Functions 393
At() 397
Atc() 397
AtStart( ) 341
Autofit 80
Autolinks 188
Average( ) 393
AverageNZ( ) 393

## - B -

Backing Up and Restoring User Data 244
Barrier Trial Groups (Daily Track Work Form) 116
Barrier Trials (Daily Track Work Form) 116
Basic Steps 220
Basic UI Techniques 72
Betting Index 218
Box( ) 437
Built-in Starter Views 168
Built-in Views 168

## - C -

Cache for View Data 149
Changing a Column's Position 83
Changing and Restoring Form Size 79
Changing Column Width 80
Changing Language of Display 72
Changing Row Height 81
Characteristics of Null Values 257
Chinese Characters Becomes Garbage 466
Chinese display 72
Clearing Cache Folder 248
Clothers 104
Cnv( ) 418
Column position 83
Column width 80
Column( ) 342
ComingDate( ) 316
Compare( ) 439
Comprehensive Statistics & Record Query 196
Comprehensive Statistics and Record Query Form 119
Connecting to the Internet 66
Conv( ) 420
Conversion Functions 418
Conversion Tables 150
Copying a Starter View 139
Create (.AND.) Bracket 209
Create (.OR.) Bracket 209
Create AND Criteria Group 209
Create OR Criteria Group 209
Creating a Report for a View 182
Creating a Starter View 130
Creating MCL Variables 160
Creating Starter Views 123
Criteria Group 209
Cross-reference 73
Current Odds from Jockey Club 115
CurRun( ) 343
CurRunHorses( ) 316

## - D -

Daily Track Work and Barrier Trial 115
Data Collection Assistant Functions 341
Data Export via Starter Views 233
Data Period and Display Mode 211
Data Range 470
Data Types of Arguments 253
Data Update 66
Date Search 74
Date( ) 440
DateOfRace( ) 318
DateStr( ) 420
Defining Double-Click Actions 117
Defining Starter Views 122
Defining Structure of Statistics Views and Queries 197
Dial-Up 67
Dividends Page (Race Data Form) 100
Downloading Data 69
Draw Position Statistics 219
DToC( ) 421

© Welkin Systems Limited

# - E -

Editing Data 77
Editing View Column/Criteria Items 204
Editing View Particulars 202
Edition 440
Empty( ) 441
English display 72
Errors in MCL Formulas 258
EvI( ) 421
Excel Spreadsheet Features 178
Export and Import View Data 158
Exporting and Importing Views 137
Exporting Contents of Grids in Forms 230
Exporting Data 228
Exporting Horse Data 229
Exporting Race Data 228
Exporting View Data 230

# - F -

Faile to Open a Starter View 467
Filter Editor 146
Filtering Past Records 136
Filtering Records in Starters Views 142
Find horses of this race met in past starts 450
Find horses which win in 60 days after a stable change 452
Firewall Blocking 70
Flemming Page (Race Data Form) 99
Format of Exported Excel Files 238
Freezing a Form 79
Freezing Columns 82
Functions 447

# - G -

GCnv( ) 422
GConv( ) 423
GeI( ) 441
GetColumn( ) 343

# - H -

Handling Forms 77
Handling Grids 79
History Tracks 139
Horse Clothes 104
Horse Form 104
Horse List Page (Horse Form) 109
Horses in Stable (Stable Form) 113
How Can I Display Chinese in English Windows 465
How To 450
How to Create Complicated Formulae 178
HrABTCount( ) 303
HrAccSectionalTime( ) 318
HrAccSectionalTimew( ) 319
HrActivity( ) 304
HrAge( ) 259
HrAgeMonths( ) 259
HrAllowance( ) 260
HrArrivalYear( ) 286
HrAverage( ) 354
HrAverageBy( ) 355
HrAverageNZ( ) 357
HrAveRating( ) 319
HrBestGallopSect( ) 307
HrBleedDate( ) 260
HrBleedDateStr( ) 260
HrBlinker( ) 260
HrBlinkerAbbr( ) 261
HrBlinkerAbbrNew( ) 261
HrBlinkerNew( ) 261
HrBodyWeight( ) 262
HrBrandNo( ) 287
HrBreedNote( ) 287
HrBTrial( ) 307
HrBTVideo( ) 431
HrCHorse( ) 287
HrClass( ) 262
HrClassGroup( ) 319
HrClassID( ) 262
HrColor( ) 288
HrCount( ) 358
HrCountBy( ) 360
HrCountNZ( ) 366
HrCountryID( ) 288
HrCourse( ) 319
HrCourseDist( ) 320
HrCourseID( ) 320
HrCupName( ) 321
HrCurRating( ) 263
HrDam( ) 288
HrData( ) 344
HrDate( ) 321
HrDayNight( ) 321
HrDayNote( ) 322
HrDaysAgo( ) 322
HrDDam( ) 289
HrDDDam( ) 289
HrDDSire( ) 289

© Welkin Systems Limited

HrDistance( ) 322
HrDistTrack( ) 323
HrDraw( ) 263
HrDSDam( ) 290
HrDSire( ) 290
HrDSSire( ), HrDSSire( ) 291
HrDvd( ) 323
HrDvdPlace( ) 324
HrDvdTrio( ) 325
HrDvdWin( ) 325
HrEHorse( ) 291
HrEntriesWeight( ) 264
HrFavorite( ) 264
HrFinalOdds( ) 265
HrFoalDate( ) 291
HrFP( ) 265
HrFPX( ) 266
HrGallop1200Best( ) 309
HrGallop1200Stat( ) 309
HrGallop400Best( ) 310
HrGallop400Stat( ) 310
HrGallop800Best( ) 310
HrGallop800Stat( ) 311
HrGBTCount( ) 311
HrGoing( ) 325
HrGoingAbbr( ) 326
HrGoingID( ) 327
HrHistoryList( ) 345
HrHorse( ) 292
HrHorseID( ) 292
HrHorseMarkNote( ) 292
HrHorseNo( ) 266
HrHorseNoStr( ) 266
HrHorseNote( ) 293
HrIsRunner( ) 267
HrIsTurf( ) 327
HrJockey( ) 293
HrJockeyClass( ) 293
HrJockeyEngAbbr( ) 294
HrJockeyID( ) 294
HrJockeyRetired( ) 294
HrLastSectionTime( ) 328
HrLastSectionTimew( ) 329
HrLastSectPos( ) 267
HrLoadDiff( ) 268
HrLoadDiffStr( ) 268
HrMargin( ) 268
HrMarginLen( ) 269
HrMarginLenX( ) 269
HrMaximum( ) 367
HrMaxStartID( ) 369
HrMeetingNo( ) 329
HrMinimum( ) 370
HrMinStartID( ) 371
HrNetLoadDiff( ) 270
HrNoteMark( ) 295
HrOdds( ) 270
HrOdds1Day( ) 270
HrOdds5Min( ) 271
HrOrigin( ) 295
HrOriginCategory( ) 295
HrOverweight( ) 271
HrOwner( ) 296
HrPastIncident( ) 271
HrPenetrometer( ) 329
HrPlaceTickets( ) 272
HrPositions( ) 272
HrPosPhoto( ) 434
HrPostRace( ) 432
HrPowerComments( ) 337
HrPowerFigure( ) 336
HrPreRace( ) 433
HrPrivilege( ) 272
HrPrivilegeID( ) 273
HrRaceClass( ) 330
HrRaceID( ) 330
HrRaceNoInMeeting( ) 330
HrRaceNoInSeason( ) 331
HrRaceNote( ) 331
HrRacePrize( ) 331
HrRaceRating( ) 332
HrRaceRemark( ) 332
HrRaceReplay( ) 435
HrRacesInMeeting( ) 332
HrRaceTime( ) 332
HrRacingFocus( ) 436
HrRacingWeight( ) 273
HrRating( ) 274
HrRatingDiff( ) 274
HrReserved( ) 274
HrRetired() 296
HrRGBTCount( ) 312
HrRunners( ) 333
HrRunning( ) 273
HrSameBlood( ) 296
HrSDam( ) 297
HrSDDam( ) 297
HrSDSire( ) 298
HrSeason( ) 333
HrSeason2( ) 334
HrSeasonRace( ) 334
HrSeconds( ) 275
HrSectionalTime( ) 335
HrSectionalTimew( ) 335

© Welkin Systems Limited

HrSectionPos( ) 275
HrSectionTime( ) 335
HrSectionTimew( ) 336
HrSectMargins( ) 275
HrSectMargLens( ) 276
HrSectMargLensX( ) 277
HrSectPositions( ) 277
HrSectTime( ) 278
HrSectTimew( ) 279
HrSex( ) 298
HrSickHistory( ) 298
HrSire( ) 300
HrSpecialBit( ) 279
HrSpecialBitAbbr( ) 279
HrSpeed( ) 280
HrSSDam( ) 301
HrSSire( ) 301
HrSSSire( ) 301
HrStable( ) 302
HrStableAbbr( ) 302
HrStableID( ) 302
HrStablePrefence( ) 280
HrStableRetired( ) 303
HrStakes( ) 281
HrStartID( ) 280
HrStartIDOn() 281
HrStatList( ) 373
HrSum( ) 374
HrSumBy( ) 375
HrSumCount( ) 377
HrTime( ) 283
HrTimew( ) 283
HrTrack( ) 337
HrTrackID( ) 337
HrTrackPlus( ) 338
HrTrackShift( ) 339
HrTwCount( ) 312
HrTwResumeDate( ) 306
HrUpdown( ) 283
HrUpdownID( ) 284
HrWeight( ) 284
HrWinner( ) 339
HrWinnerID( ) 339
HrWinSeconds( ) 340
HrWinTickets( ) 285
HrWinTime( ) 340
HrWinTimew( ) 340
HrWQPStat( ) 285
HrXAverage( ) 379
HrXMaximum( ) 380
HrXMaxStartID( ) 381
HrXMinimum( ) 382
HrXMinStartID( ) 383
HrXSum( ) 385
Hyper-Linking 73

## - I -

ICase 442
IfAtTop( ) 346
IfCurRun( ) 347
IIF( ) 442
Importing View Data From Excel Files 240
Incremental Search 74
Index Page (Race Data Form) 101
Information about a Start 258
InList( ) 443
Installing RaceMate 21
Int( ) 394
Introduction to RaceMate 18
IrregularRace( ) 341
IsNull( ) 443

## - J -

Jockey Change Statistics 171
Jockey Form 109
Jockeys' Clothes 104
Jockeys List Page (Jockey Form) 112
Join( ) 443
JrABTCount( ) 303
JrAccSectionalTime( ) 318
JrAccSectionalTimew( ) 319
JrActivity( ) 304
JrAge( ) 259
JrAgeMonths( ) 259
JrAllowance( ) 260
JrArrivalYear( ) 286
JrAverage( ) 354
JrAverageBy( ) 355
JrAverageNZ( ) 357
JrAveRating( ) 319
JrBestGallopSect( ) 307
JrBleedDate( ) 260
JrBleedDateStr( ) 260
JrBlinker( ) 260
JrBlinkerAbbr( ) 261
JrBlinkerAbbrNew( ) 261
JrBlinkerNew( ) 261
JrBodyWeight( ) 262
JrBrandNo( ) 287
JrBreedNote( ) 287
JrBTrial( ) 307

© Welkin Systems Limited

JrBTVideo( ) 431
JrCHorse( ) 287
JrClass( ) 262
JrClassGroup( ) 319
JrClassID( ) 262
JrColor( ) 288
JrCount( ) 358
JrCountBy( ) 360
JrCountNZ( ) 366
JrCountryID( ) 288
JrCourse( ) 319
JrCourseDist( ) 320
JrCourseID( ) 320
JrCupName( ) 321
JrCurRating( ) 263
JrDam( ) 288
JrData( ) 344
JrDate( ) 321
JrDayNight( ) 321
JrDayNote( ) 322
JrDaysAgo( ) 322
JrDDam( ) 289
JrDDDam( ) 289
JrDDSire( ) 289
JrDistance( ) 322
JrDistTrack( ) 323
JrDraw( ) 263
JrDSDam( ) 290
JrDSire( ) 290
JrDvd( ) 323
JrDvdPlace( ) 324
JrDvdTrio( ) 325
JrDvdWin( ) 325
JrEHorse( ) 291
JrEntriesWeight( ) 264
JrFavorite( ) 264
JrFinalOdds( ) 265
JrFoalDate( ) 291
JrFP( ) 265
JrFPX( ) 266
JrGallop1200Best( ) 309
JrGallop1200Stat( ) 309
JrGallop400Best( ) 310
JrGallop400Stat( ) 310
JrGallop800Best( ) 310
JrGallop800Stat( ) 311
JrGBTCount( ) 311
JrGoing( ) 325
JrGoingAbbr( ) 326
JrGoingID( ) 327
JrHistoryList( ) 345
JrHorse( ) 292
JrHorseID( ) 292
JrHorseMarkNote( ) 292
JrHorseNo( ) 266
JrHorseNoStr( ) 266
JrHorseNote( ) 293
JrIsRunner( ) 267
JrIsTurf( ) 327
JrJockey( ) 293
JrJockeyClass( ) 293
JrJockeyEngAbbr( ) 294
JrJockeyID( ) 294
JrJockeyRetired( ) 294
JrLastSectionTime( ) 328
JrLastSectionTimew( ) 329
JrLastSectPos( ) 267
JrLoadDiff( ) 268
JrLoadDiffStr( ) 268
JrMargin( ) 268
JrMarginLen( ) 269
JrMarginLenX( ) 269
JrMaximum( ) 367
JrMaxStartID( ) 369
JrMeetingNo( ) 329
JrMinimum( ) 370
JrMinStartID( ) 371
JrNetLoadDiff( ) 270
JrNoteMark( ) 295
JrOdds( ) 270
JrOdds1Day( ) 270
JrOdds5Min( ) 271
JrOrigin( ) 295
JrOriginCategory( ) 295
JrOverweight( ) 271
JrOwner( ) 296
JrPastIncident( ) 271
JrPenetrometer( ) 329
JrPlaceTickets( ) 272
JrPositions( ) 272
JrPosPhoto( ) 434
JrPostRace( ) 432
JrPowerComments( ) 337
JrPowerFigure( ) 336
JrPreRace( ) 433
JrPrivilege( ) 272
JrPrivilegeID( ) 273
JrRaceClass( ) 330
JrRaceID( ) 330
JrRaceNoInMeeting( ) 330
JrRaceNoInSeason( ) 331
JrRaceNote( ) 331
JrRacePrize( ) 331
JrRaceRating( ) 332

© Welkin Systems Limited

JrRaceRemark( ) 332
JrRaceReplay( ) 435
JrRacesInMeeting( ) 332
JrRaceTime( ) 332
JrRacingFocus( ) 436
JrRacingWeight( ) 273
JrRating( ) 274
JrRatingDiff( ) 274
JrReserved( ) 274
JrRetired() 296
JrRGBTCount( ) 312
JrRunners( ) 333
JrRunning( ) 273
JrSameBlood( ) 296
JrSDam( ) 297
JrSDDam( ) 297
JrSDSire( ) 298
JrSeason( ) 333
JrSeason2( ) 334
JrSeasonRace( ) 334
JrSeconds( ) 275
JrSectionalTime( ) 335
JrSectionalTimew( ) 335
JrSectionPos( ) 275
JrSectionTime( ) 335
JrSectionTimew( ) 336
JrSectMargins( ) 275
JrSectMargLens( ) 276
JrSectMargLensX( ) 277
JrSectPositions( ) 277
JrSectTime( ) 278
JrSectTimew( ) 279
JrSex( ) 298
JrSickHistory( ) 298
JrSire( ) 300
JrSpecialBit( ) 279
JrSpecialBitAbbr( ) 279
JrSpeed( ) 280
JrSSDam( ) 301
JrSSire( ) 301
JrSSSire( ) 301
JrStable( ) 302
JrStableAbbr( ) 302
JrStableID( ) 302
JrStablePrefence( ) 280
JrStableRetired( ) 303
JrStakes( ) 281
JrStartID( ) 280
JrStartIDOn() 281
JrStatList( ) 373
JrSum( ) 374
JrSumBy( ) 375
JrSumCount( ) 377
JrTime( ) 283
JrTimew( ) 283
JrTrack( ) 337
JrTrackID( ) 337
JrTrackPlus( ) 338
JrTrackShift( ) 339
JrTwCount( ) 312
JrTwResumeDate( ) 306
JrUpdown( ) 283
JrUpdownID( ) 284
JrWeight( ) 284
JrWinner( ) 339
JrWinnerID( ) 339
JrWinSeconds( ) 340
JrWinTickets( ) 285
JrWinTime( ) 340
JrWinTimew( ) 340
JrWQPStat( ) 285
JrXAverage( ) 379
JrXMaximum( ) 380
JrXMaxStartID( ) 381
JrXMinimum( ) 382
JrXMinStartID( ) 383
JrXSum( ) 385
Jump by Double-Clicking a Column 117

# - L -

Lang( ) 444
Language of display 72
Left( ) 397
License Agreement 474
Like( ) 398
Linked Columns 155
Linking Up Views 155
Links Page (Daily Track Work Form) 117
Links Page (Horse Form) 108
Links Page (Jockey Form) 111
Links Page (Race Data Form) 101
Links Page (Stable Form) 114
List Functions 404
ListAscItem( ) 404
ListAscVal( ) 407
ListAverage( ) 407
ListCol( ) 408
ListColVal( ) 408
ListCommon( ) 408
ListCount( ) 408
ListDescItem( ) 409
ListDescVal( ) 410

© Welkin Systems Limited

ListFilter( ) 410
ListItem( ) 411
ListSort( ) 412
ListSortDesc( ) 414
ListsortDescVal( ) 414
ListSortVal( ) 414
ListStdDev( ) 415
ListSum( ) 415
ListVal( ) 416
LTrim( ) 398

# - M -

Main Data Forms 90
Main Features 18
Managing Conversion Tables 151
Managing Starter Views 122
Managing Your Own Multimedia Resources 189
Match( ) 444
Maximum( ) 394
MCL 252
MCL Condition 254
MCL Debugging 178
MCL Reference 252
MCL Tutorial 250
MCL Variable 160
MCL Variables in MCL Functions 160
MCL Variables to Set Data Range 164
Menus and Toolbars 72
Minimum( ) 395
Miscellaneous Functions 437
Model Construction Language 252
Modifying Criteria and Setting Compound Filters 143
Modifying Existing Views 127
Month( ) 445
Moving or Uninstalling RaceMate 28
Moving Toolbars 72
Multimedia & Internet Functions 431
Multimedia Resources 188
Multiple instances 73
Multiple Instances of a Form 78

# - N -

Notes Page (Jockey Form) 111
Notes Page (Stable Form) 114
Null Values 257
Null Values in Statistics 257
Number Search 74
NVL( ) 425

# - O -

Occurs( ) 399
Operators 448
Optional Arguments 253
Options and System Information 246
Over( ) 395
Overseas Page (Horse Form) 107

# - P -

Padr( ) 400
Particulars Page (Horse Form) 105
Past Record and Comprehensive Statistics (Stable Form) 113
Past Record of Horse 106
Past Record of Jockey 109
Pedigree 21
Pedigree Page (Horse Form) 106
Prediction Models 218
Print Race Data (Race Data Form) 102
Printing and Creating Reports 182
Printing or Exporting Statistics 224
Propagation of Null Values 257

# - Q -

QStr( ) 424
Query Operators for Requirements 206
Questions & Answers 464
Questions You May Asked After Upgrading to 9.0 464
Quoted MCL Expression Arguments 255

# - R -

Race and Meeting Information 314
Race Calendar Form 94
Race Data Form 95
Race Search Form 103
RaceMate Freezed When Starter View Being Opened 467
RaceMate Notice 245
RaceVar( ) 348
Rearranging Position of Views in View List 136
Record ID 119
Record Tree Form 90
Registering and Activating RaceMate 23
Release Notes 36
Remark Page (Race Data Form) 100

© Welkin Systems Limited

Remarks about Inputting Chinese 33
Repeat Last Download 69
Replicate( ) 400
Report 102
Report Designer 183
Resetting Grid Layout 86
Resizing Columns 80
Restore Form Size 79
Right( ) 401
RListItem( ) 417
RListVal( ) 417
Row ID 119
RunnersAvg( ) 387
RunnersCnt( ) 388
RunnersFind( ) 389
RunnersList( ) 390
RunnersMax( ) 390
RunnersMin( ) 391
RunnersSum( ) 392

# - S -

Same( ) 446
Saving Web Video/Photo into Disk 191
SCnv( ) 425
SConv( ) 426
Season( ) 446
See performance enhanced by a stable change 457
Selecting Criteria 142
Setting Data Range in an Argument 149
Setting Values & Choosing MCL Variables 142, 161
Show Starters View History 139
Simple Statistics of Jockey 110
Simple Statistics of Stable 113
Simplified Statistics Form 225
Sorting and Grouping 141
Sorting Grid Columns 84
Sorting in Individual History Groups 141
Sorting in statistics views 201
SrABTCount( ) 303
SrAccSectionalTime( ) 318
SrAccSectionalTimew( ) 319
SrActivity( ) 304
SrAge( ) 259
SrAgeMonths( ) 259
SrAllowance( ) 260
SrArrivalYear( ) 286
SrAverage( ) 354
SrAverageBy( ) 355
SrAverageNZ( ) 357
SrAveRating( ) 319
SrBestGallopSect( ) 307
SrBleedDate( ) 260
SrBleedDateStr( ) 260
SrBlinker( ) 260
SrBlinkerAbbr( ) 261
SrBlinkerAbbrNew( ) 261
SrBlinkerNew( ) 261
SrBodyWeight( ) 262
SrBrandNo( ) 287
SrBreedNote( ) 287
SrBTrail( ) 307
SrBTVideo( ) 431
SrCHorse( ) 287
SrClass( ) 262
SrClassGroup( ) 319
SrClassID( ) 262
SrColor( ) 288
SrCount( ) 358
SrCountBy( ) 360
SrCountNZ( ) 366
SrCountryID( ) 288
SrCourse( ) 319
SrCourseDist( ) 320
SrCourseID( ) 320
SrCupName( ) 321
SrCurRating( ) 263
SrDam( ) 288
SrData( ) 344
SrDate( ) 321
SrDayNight( ) 321
SrDayNote( ) 322
SrDaysAgo( ) 322
SrDDam( ) 289
SrDDDam( ) 289
SrDDSire( ) 289
SrDistance( ) 322
SrDistTrack( ) 323
SrDraw( ) 263
SrDSDam( ) 290
SrDSire( ) 290
SrDSSire( ) 291
SrDvd( ) 323
SrDvdPlace( ) 324
SrDvdTrio( ) 325
SrDvdWin( ) 325
SrEHorse( ) 291
SrEntriesWeight( ) 264
SrFavorite( ) 264
SrFinalOdds( ) 265
SrFoalDate( ) 291
SrFP( ) 265
SrFPX( ) 266

© Welkin Systems Limited

SrGallop1200Best( ) 309
SrGallop1200Stat( ) 309
SrGallop400Best( ) 310
SrGallop400Stat( ) 310
SrGallop800Best( ) 310
SrGallop800Stat( ) 311
SrGBTCount( ) 311
SrGoing( ) 325
SrGoingID( ) 327
SrGroingAbbr( ) 326
SrHistoryList( ) 345
SrHorse( ) 292
SrHorseID( ) 292
SrHorseMarkNote( ) 292
SrHorseNo( ) 266
SrHorseNoStr( ) 266
SrHorseNote( ) 293
SrIsRunner( ) 267
SrIsTurf( ) 327
SrJockey( ) 293
SrJockeyClass( ) 293
SrJockeyEngAbbr( ) 294
SrJockeyID( ) 294
SrJockeyRetired( ) 294
SrLastSectionTime( ) 328
SrLastSectionTimew( ) 329
SrLastSectPos( ) 267
SrLoadDiff( ) 268
SrLoadDiffStr( ) 268
SrMargin( ) 268
SrMarginLen( ) 269
SrMarginLenX( ) 269
SrMaxmum( ) 367
SrMaxStartID( ) 369
SrMeetingNo( ) 329
SrMinimum( ) 370
SrMinStartID( ) 371
SrNetLoadDiff( ) 270
SrNoteMark( ) 295
SrOdds( ) 270
SrOdds1Day( ) 270
SrOdds5Min( ) 271
SrOrigin( ) 295
SrOriginCategory( ) 295
SrOverweight( ) 271
SrOwner( ) 296
SrPenetrometer( ) 329
SrPlaceTickets( ) 272
SrPositions( ) 272
SrPosPhoto( ) 434
SrPostRace( ) 432
SrPowerComments( ) 337
SrPowerFigure( ) 336
SrPreRace( ) 433
SrPrivilege( ) 272
SrPrivilegeID( ) 273
SrRaceClass( ) 330
SrRaceID( ) 330
SrRaceNoInMeeting( ) 330
SrRaceNoInSeason( ) 331
SrRaceNote( ) 331
SrRacePrize( ) 331
SrRaceRating( ) 332
SrRaceRemark( ) 332
SrRaceReplay( ) 435
SrRacesInMeeting( ) 332
SrRaceTime( ) 332
SrRacingFocus( ) 436
SrRacingWeight( ) 273
SrRating( ) 274
SrRatingDiff( ) 274
SrReserved( ) 274
SrRetired() 296
SrRGBTCount( ) 312
SrRunners( ) 333
SrRunning( ) 273
SrSameBlood( ) 296
SrSDam( ) 297
SrSDDam( ) 297
SrSDSire( ) 298
SrSeason( ) 333
SrSeason2( ) 334
SrSeasonRace( ) 334
SrSeconds( ) 275
SrSectionalTime( ) 335
SrSectionalTimew( ) 335
SrSectionPos( ) 275
SrSectionTime( ) 335
SrSectionTimew( ) 336
SrSectMargins( ) 275
SrSectMargLens( ) 276
SrSectMargLensX( ) 277
SrSectPositions( ) 277
SrSectTime( ) 278
SrSectTimew( ) 279
SrSex( ) 298
SrSickHistory( ) 298
SrSire( ) 300
SrSpecialBit( ) 279
SrSpecialBitAbbr( ) 279
SrSpeed( ) 280
SrSSDam( ) 301
SrSSire( ) 301
SrSSSire( ) 301

© Welkin Systems Limited

SrStable( ) 302
SrStableAbbr( ) 302
SrStableID( ) 302
SrStablePrefence( ) 280
SrStableRetired( ) 303
SrStakes( ) 281
SrStartID( ) 280
SrStartIDOn() 281
SrStatList( ) 373
SrSum( ) 374
SrSumBy( ) 375
SrSumCount( ) 377
SrTime( ) 283
SrTimew( ) 283
SrTrack( ) 337
SrTrackID( ) 337
SrTrackPlus( ) 338
SrTrackShift( ) 339
SrTwCount( ) 312
SrTwResumeDate( ) 306
SrUpdown( ) 283
SrUpdownID( ) 284
SrWeight( ) 284
SrWinner( ) 339
SrWinnerID( ) 339
SrWinSeconds( ) 340
SrWinTickets( ) 285
SrWinTime( ) 340
SrWinTimew( ) 340
SrWQPStat( ) 285
SrXAverage( ) 379
SrXMaximum( ) 380
SrXMaxStartID( ) 381
SrXMinimum( ) 382
SrXMinStartID( ) 383
SrXSum( ) 385
Stable Form 112
Stables List Page (Stable Form) 115
Starter View Page (Race Data Form) 96
StartVar( ) 351
Stat Column 214
Static information of Horse, Jockey, and Stable 285
Statistical Groups 222
Statistical Items 214
Statistical View and Record Filtering Results 201
Statistics and Data Analysis 196
Statistics in a Race 387
Statistics on Past Records 353
Statistics via Starter Views (Race Data Form) 218
StdTime( ) 386
StdTimew( ) 387
SToT( ) 427
Str( ) 427
StrDiff( ) 428
StrExtract( ) 401
String Functions 396
SubStr( ) 402
Sum( ) 396
System Configurations and Information 246
System Maintenance and Configuration 244
System Requirements 20

# - T -

TblAddVal( ) 428
TblWrite( ) 429
Technical Reference 470
Text Search 74
The cIncluding Argument 256
The DefaultVal Argument 256
The nBackRaces Argument of an MCL function 254
Track Work (Daily Track Work Form) 116
Track Work Information 303
Trackwork of Horse 108
Trackwork Page (Race Data Form) 97
Transform( ) 430
Trim( ) 403
TToS( ) 430
Type 1: Starter Views 122
Type 2: Histroy and Statistics View 148
Types of History and Statistics Views 148

# - U -

Upgrading RaceMate or Installing Add-Ins 248
User ID 23
Using Conversion Tables in Views 153
Using Menus and Toolbars 72

# - V -

Val( ) 431
Variable 160
VFP Functions 447
VFP Functions & Operators 447
VFP Operators 448
View Header 128
View Management 122
Viewing Past Records and Move Columns in Stat View Page 216

© Welkin Systems Limited

# - W -

What is New in RaceMate 10.0 38
What is New in RaceMate 10.5 37
What is New in RaceMate 11.5 36
What is New in RaceMate 7.0 59
What is New in RaceMate 8.0 50
What is New in RaceMate 9.0 47
What is New in RaceMate 9.5 43
What is New in RaceMate 9.8 40
When to use the While keyword 255
Wins Page (Horse Form) 107

# - Y -

Year( ) 447

© Welkin Systems Limited